Sterling Commerce
An IBM Company

Selling and Fulfillment Foundation: Using the Rich Client Platform Extensibility Tool Guide

Release 8.5

October 2009

Copyright Notice

STERLING COMMERCE SOFTWARE

Sterling Commerce, Inc.
4600 Lakehurst Court
Dublin, Ohio 43016-2000

Third-Party Software

Portions of the Sterling Commerce Software may include products, or may be distributed on the same storage media with products, ("Third Party Software") offered by third parties ("Third Party Licensors"). Sterling Commerce Software may include Third Party Software covered by the following copyrights:

The Sterling Commerce Software is distributed on the same storage media as certain Third Party Software covered by the following copyrights:

Certain components of the Sterling Commerce Software are distributed on the same storage media as Third Party Software which are not listed above. Additional information for such Third Party Software components of the Sterling Commerce Software is located at: installdir/mesa/studio/plugins/SCI_Studio_License.txt.

Third Party Software which is included, or are distributed on the same storage media with, the Sterling Commerce Software where use, duplication, or disclosure by the United States government or a government contractor or subcontractor, are provided with RESTRICTED RIGHTS under Title 48 CFR 2.101, 12.212, 52.227-19, 227.7201 through 227.7202-4, DFAR 252.227-7013(c) (1) (ii) and (2), DFAR 252.227-7015(b)(6/95), DFAR 227.7202-3(a), FAR 52.227-14(g)(2)(6/87), and FAR 52.227-19(c)(2) and (6/87) as applicable.

Additional information regarding certain Third Party Software is located at installdir/SCI_License.txt.

Some Third Party Licensors also provide license information and/or source code for their software via their respective links set forth below:

http://danadler.com/jacob/

http://www.dom4j.org

This product includes software developed by the Apache Software Foundation (http://www.apache.org). This product includes software developed by the Ant-Contrib project (http://sourceforge.net/projects/ant-contrib). This product includes software developed by the JDOM Project (http://www.jdom.org/). This product includes code licensed from RSA Data Security (via Sun Microsystems, Inc.). Sun, Sun Microsystems, the Sun Logo, Java, JDK, the Java Coffee Cup logo, JavaBeans , JDBC, JMX and all JMX based trademarks and logos are trademarks or registered trademarks of Sun Microsystems, Inc. All other trademarks and logos are trademarks of their respective owners.

THE APACHE SOFTWARE FOUNDATION SOFTWARE

The Sterling Commerce Software is distributed with or on the same storage media as the following software products (or components thereof) and java source code files: Xalan version 2.5.2, Cookie.java, Header.java, HeaderElement.java, HttpException.java, HttpState.java, NameValuePair.java, CronTimeTrigger.java, DefaultTimeScheduler.java, PeriodicTimeTrigger.java, Target.java, TimeScheduledEntry.java, TimeScheduler.java, TimeTrigger.java, Trigger.java, BinaryHeap.java, PriorityQueue.java, SynchronizedPriorityQueue.java, GetOpt.java, GetOptsException.java, IllegalArgumentException.java, MissingOptArgException.java (collectively, "Apache 1.1 Software"). Apache 1.1 Software is free software which is distributed under the terms of the following license:

License Version 1.1

Copyright 1999-2003 The Apache Software Foundation. All rights reserved.

Redistribution and use in source and binary forms, with or without modification, are permitted provided that the following conditions are met:

1. Redistributions of source code must retain the above copyright notice, this list of conditions and the following disclaimer.

2. Redistribution in binary form must reproduce the above copyright notice, this list of conditions and the following disclaimer in the documentation and/or other materials provided with the distribution.

3. The end-user documentation included with the redistribution, if any, must include the following acknowledgement: "This product includes software developed by the Apache Software Foundation (http://www.apache.org)." Alternatively, this acknowledgement may appear in the software itself, if and whenever such third-party acknowledgements normally appear.

4. The names "Commons", "Jakarta", "The Jakarta Project", "HttpClient", "log4j", "Xerces "Xalan", "Avalon", "Apache Avalon", "Avalon Cornerstone", "Avalon Framework", "Apache" and "Apache Software Foundation" must not be used to endorse or promote products derived from this software without specific prior written permission. For written permission, please contact apache@apache.org.

5. Products derived from this software may not be called "Apache", nor may "Apache" appear in their name, without the prior written permission of the Apache Software Foundation.

THIS SOFTWARE IS PROVIDED "AS IS" AND ANY EXPRESS OR IMIPLIED WARRANTIES, INCLUDING ANY IMPLIED WARRANTY OF MERCHANTIBILITY, AND FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL APACHE SOFTWARE FOUNDATION OR ITS CONTRIBUTORS BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTIAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

This software consists of voluntary contributions made by many individuals on behalf of the Apache Software Foundation. The GetOpt.java, GetOptsException.java, IlligalArgumentException.java and MissingOptArgException.java software was originally based on software copyright (c) 2001, Sun Microsystems., http://www.sun.com. For more information on the Apache Software Foundation, please see <http://www.apache.org/>.

The preceding license only applies to the Apache 1.1 Software and does not apply to the Sterling Commerce Software or to any other Third-Party Software.

The Sterling Commerce Software is also distributed with or on the same storage media as the following software products (or components thereof): Ant, Antinstaller, Apache File Upload Package, Apache Commons Beans, Apache Commons BetWixt, Apache Commons Collection, Apache Commons Digester, Apache Commons IO, Apache Commons Lang., Apache Commons Logging, Apache Commons Net, Apache Jakarta Commons Pool, Apache Jakarta ORO, Lucene, Xerces version 2.7, Apache Log4J, Apache SOAP, Apache Struts and Apache Xalan 2.7.0, (collectively, "Apache 2.0 Software"). Apache 2.0 Software is free software which is distributed under the terms of the Apache License Version 2.0. A copy of License Version 2.0 is found in the following directory files for the individual pieces of the Apache 2.0 Software: installdir/jar/commons_upload/1_0/ CommonsFileUpload_License.txt,
installdir/jar/jetspeed/1_4/RegExp_License.txt,
installdir/ant/Ant_License.txt
<install>/jar/antInstaller/0_8/antinstaller_License.txt,
<install>/jar/commons_beanutils/1_7_0/commons-beanutils.jar (/META-INF/LICENSE.txt),
<install>/jar/commons_betwixt/0_8/commons-betwixt-0.8.jar (/META-INF/LICENSE.txt),

<install>/jar/commons_collections/3_2/LICENSE.txt,
<install>/jar/commons_digester/1_8/commons-digester-1.8.jar (/META-INF/LICENSE.txt),
<install>/jar/commons_io/1_4/LICENSE.txt,
<install>/jar/commons_lang/2_1/Commons_Lang_License.txt,
<install>/jar/commons_logging/1_0_4/commons-logging-1.0.4.jar (/META-INF/LICENSE.txt),
<install>/jar/commons_net/1_4_1/commons-net-1.4.1.jar (/META-INF/LICENSE.txt),
<install>/jar/smcfs/8.5/lucene-core-2.4.0.jar (/META-INF/LICENSE.txt),
<install>/jar/struts/2_0_11/struts2-core-2.0.11.jar (./LICENSE.txt),
<install>/jar/mesa/gisdav/WEB-INF/lib/Slide_License.txt,
<install>/mesa/studio/plugins/xerces_2.7_license.txt,
<install>/jar/commons_pool/1_2/Commons_License.txt,
<install>/jar/jakarta_oro/2_0_8/JakartaOro_License.txt,
<install>/jar/log4j/1_2_15/LOG4J_License.txt,
<install>/jar/xalan/2_7/Xalan_License.txt,
<install>/jar/soap/2_3_1/Apache_SOAP_License.txt

Unless otherwise stated in a specific directory, the Apache 2.0 Software was not modified. Neither the Sterling Commerce Software, modifications, if any, to Apache 2.0 Software, nor other Third Party Code is a Derivative Work or a Contribution as defined in License Version 2.0. License Version 2.0 applies only to the Apache 2.0 Software which is the subject of the specific directory file and does not apply to the Sterling Commerce Software or to any other Third Party Software. License Version 2.0 includes the following provision:

"Unless required by applicable law or agreed to in writing, Licensor provides the Work (and each Contributor provides its Contributions) on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied, including, without limitation, any warranties or conditions of TITLE, NON-INFRINGEMENT, MERCHANTABILITY, or FITNESS FOR A PARTICULAR PURPOSE. You are solely responsible for determining the appropriateness of using or redistributing the Work and assume any risks associated with Your exercise of permissions under this License."

NOTICE file corresponding to the section 4 d of the Apache License, Version 2.0, in this case for the Apache Ant distribution. Apache Ant Copyright 1999-2008 The Apache Software Foundation. This product includes software developed by The Apache Software Foundation (http://www.apache.org/). This product includes also software developed by :

- the W3C consortium (http://www.w3c.org) ,

- the SAX project (http://www.saxproject.org)

The <sync> task is based on code Copyright (c) 2002, Landmark Graphics Corp that has been kindly donated to the Apache Software Foundation.

Portions of this software were originally based on the following:

- software copyright (c) 1999, IBM Corporation., http://www.ibm.com.

- software copyright (c) 1999, Sun Microsystems., http://www.sun.com.

- voluntary contributions made by Paul Eng on behalf of the Apache Software Foundation that were originally developed at iClick, Inc., software copyright (c) 1999.

NOTICE file corresponding to the section 4 d of the Apache License, Version 2.0, in this case for the Apache Lucene distribution. Apache Lucene Copyright 2006 The Apache Software Foundation. This product includes software developed by The Apache Software Foundation (http://www.apache.org/). The snowball stemmers in contrib/snowball/src/java/net/sf/snowball were developed by Martin Porter and Richard Boulton. The full snowball package is available from http://snowball.tartarus.org/

Ant-Contrib Software

The Sterling Commerce Software is distributed with or on the same storage media as the Anti-Contrib software (Copyright (c) 2001-2003 Ant-Contrib project. All rights reserved.) (the "Ant-Contrib Software"). The Ant-Contrib Software is free software which is distributed under the terms of the following license:

The Apache Software License, Version 1.1

Copyright (c) 2001-2003 Ant-Contrib project. All rights reserved.

Redistribution and use in source and binary forms, with or without modification, are permitted provided that the following conditions are met:

1. Redistributions of source code must retain the above copyright notice, this list of conditions and the following disclaimer.

2. Redistributions in binary form must reproduce the above copyright notice, this list of conditions and the following disclaimer in the documentation and/or other materials provided with the distribution.

3. The end-user documentation included with the redistribution, if any, must include the following acknowledgement:

"This product includes software developed by the Ant-Contrib project (http://sourceforge.net/projects/ant-contrib)."

Alternately, this acknowledgement may appear in the software itself, if and wherever such third-party acknowledgements normally appear.

4. The name Ant-Contrib must not be used to endorse or promote products derived from this software without prior written permission. For written permission, please contact ant-contrib-developers@lists.sourceforge.net.

5. Products derived from this software may not be called "Ant-Contrib" nor may "Ant-Contrib" appear in their names without prior written permission of the Ant-Contrib project.

THIS SOFTWARE IS PROVIDED ``AS IS'' AND ANY EXPRESSED OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL THE ANT-CONTRIB PROJECT OR ITS CONTRIBUTORS BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE. The preceding license only applies to the Ant-Contrib Software and does not apply to the Sterling Commerce Software or to any other Third-Party Software.

The preceding license only applies to the Ant-Contrib Software and does not apply to the Sterling Commerce Software or to any other Third Party Software.

DOM4J Software

The Sterling Commerce Software is distributed with or on the same storage media as the Dom4h Software which is free software distributed under the terms of the following license:

Redistribution and use of this software and associated documentation ("Software"), with or without modification, are permitted provided that the following conditions are met:

1. Redistributions of source code must retain copyright statements and notices. Redistributions must also contain a copy of this document.

2. Redistributions in binary form must reproduce the above copyright notice, this list of conditions and the following disclaimer in the documentation and/or other materials provided with the distribution.

3. The name "DOM4J" must not be used to endorse or promote products derived from this Software without prior written permission of MetaStuff, Ltd. For written permission, please contact dom4j-info@metastuff.com.

4. Products derived from this Software may not be called "DOM4J" nor may "DOM4J" appear in their names without prior written permission of MetaStuff, Ltd. DOM4J is a registered trademark of MetaStuff, Ltd.

5. Due credit should be given to the DOM4J Project - http://www.dom4j.org

The preceding license only applies to the Dom4j Software and does not apply to the Sterling Commerce Software, or any other Third-Party Software.

THE ECLIPSE SOFTWARE FOUNDATION

The Sterling Commerce Software is also distributed with or on the same storage media as the following software:

com.ibm.icu.nl1_3.4.4.v200606220026.jar, org.eclipse.ant.core.nl1_3.1.100.v200606220026.jar, org.eclipse.ant.ui.nl1_3.2.0.v200606220026.jar, org.eclipse.compare.nl1_3.2.0.v200606220026.jar, org.eclipse.core.boot.nl1_3.1.100.v200606220026.jar, org.eclipse.core.commands.nl1_3.2.0.v200606220026.jar, org.eclipse.core.contenttype.nl1_3.2.0.v200606220026.jar, org.eclipse.core.expressions.nl1_3.2.0.v200606220026.jar, org.eclipse.core.filebuffers.nl1_3.2.0.v200606220026.jar, org.eclipse.core.filesystem.nl1_1.0.0.v200606220026.jar, org.eclipse.core.jobs.nl1_3.2.0.v200606220026.jar, org.eclipse.core.resources.nl1_3.2.0.v200606220026.jar, org.eclipse.core.runtime.compatibility.auth.nl1_3.2.0.v200606220026.jar, org.eclipse.core.runtime.compatibility.nl1_3.1.100.v200606220026.jar, org.eclipse.core.runtime.nl1_3.2.0.v200606220026.jar, org.eclipse.core.variables.nl1_3.1.100.v200606220026.jar, org.eclipse.debug.core.nl1_3.2.0.v200606220026.jar, org.eclipse.debug.ui.nl1_3.2.0.v200606220026.jar, org.eclipse.equinox.common.nl1_3.2.0.v200606220026.jar, org.eclipse.equinox.preferences.nl1_3.2.0.v200606220026.jar, org.eclipse.equinox.registry.nl1_3.2.0.v200606220026.jar, org.eclipse.help.appserver.nl1_3.1.100.v200606220026.jar, org.eclipse.help.base.nl1_3.2.0.v200606220026.jar, org.eclipse.help.nl1_3.2.0.v200606220026.jar, org.eclipse.help.ui.nl1_3.2.0.v200606220026.jar, org.eclipse.jdt.apt.core.nl1_3.2.0.v200606220026.jar, org.eclipse.jdt.apt.ui.nl1_3.2.0.v200606220026.jar, org.eclipse.jdt.core.manipulation.nl1_1.0.0.v200606220026.jar, org.eclipse.jdt.core.nl1_3.2.0.v200606220026.jar, org.eclipse.jdt.debug.ui.nl1_3.2.0.v200606220026.jar, org.eclipse.jdt.doc.isv.nl1_3.2.0.v200606220026.jar, org.eclipse.jdt.doc.user.nl1_3.2.0.v200606220026.jar, org.eclipse.jdt.junit4.runtime.nl1_1.0.0.v200606220026.jar, org.eclipse.jdt.launching.nl1_3.2.0.v200606220026.jar, org.eclipse.jdt.nl1_3.2.0.v200606220026.jar, org.eclipse.jdt.ui.nl1_3.2.0.v200606220026.jar, org.eclipse.jface.databinding.nl1_1.0.0.v200606220026.jar, org.eclipse.jface.nl1_3.2.0.v200606220026.jar, org.eclipse.jface.text.nl1_3.2.0.v200606220026.jar, org.eclipse.ltk.core.refactoring.nl1_3.2.0.v200606220026.jar, org.eclipse.ltk.ui.refactoring.nl1_3.2.0.v200606220026.jar, org.eclipse.osgi.nl1_3.2.0.v200606220026.jar, org.eclipse.osgi.services.nl1_3.1.100.v200606220026.jar, org.eclipse.osgi.util.nl1_3.1.100.v200606220026.jar, org.eclipse.pde.core.nl1_3.2.0.v200606220026.jar, org.eclipse.pde.doc.user.nl1_3.2.0.v200606220026.jar, org.eclipse.pde.junit.runtime.nl1_3.2.0.v200606220026.jar, org.eclipse.pde.nl1_3.2.0.v200606220026.jar, org.eclipse.pde.runtime.nl1_3.2.0.v200606220026.jar, org.eclipse.pde.ui.nl1_3.2.0.v200606220026.jar, org.eclipse.platform.doc.isv.nl1_3.2.0.v200606220026.jar, org.eclipse.platform.doc.user.nl1_3.2.0.v200606220026.jar,

org.eclipse.rcp.nl1_3.2.0.v200606220026.jar, org.eclipse.search.nl1_3.2.0.v200606220026.jar,
org.eclipse.swt.nl1_3.2.0.v200606220026.jar, org.eclipse.team.core.nl1_3.2.0.v200606220026.jar,
org.eclipse.team.cvs.core.nl1_3.2.0.v200606220026.jar,
org.eclipse.team.cvs.ssh.nl1_3.2.0.v200606220026.jar,
org.eclipse.team.cvs.ssh2.nl1_3.2.0.v200606220026.jar,
org.eclipse.team.cvs.ui.nl1_3.2.0.v200606220026.jar, org.eclipse.team.ui.nl1_3.2.0.v200606220026.jar,
org.eclipse.text.nl1_3.2.0.v200606220026.jar, org.eclipse.ui.browser.nl1_3.2.0.v200606220026.jar,
org.eclipse.ui.cheatsheets.nl1_3.2.0.v200606220026.jar,
org.eclipse.ui.console.nl1_3.1.100.v200606220026.jar,
org.eclipse.ui.editors.nl1_3.2.0.v200606220026.jar,
org.eclipse.ui.externaltools.nl1_3.1.100.v200606220026.jar,
org.eclipse.ui.forms.nl1_3.2.0.v200606220026.jar, org.eclipse.ui.ide.nl1_3.2.0.v200606220026.jar,
org.eclipse.ui.intro.nl1_3.2.0.v200606220026.jar, org.eclipse.ui.navigator.nl1_3.2.0.v200606220026.jar,
org.eclipse.ui.navigator.resources.nl1_3.2.0.v200606220026.jar,
org.eclipse.ui.nl1_3.2.0.v200606220026.jar,
org.eclipse.ui.presentations.r21.nl1_3.2.0.v200606220026.jar,
org.eclipse.ui.views.nl1_3.2.0.v200606220026.jar,
org.eclipse.ui.views.properties.tabbed.nl1_3.2.0.v200606220026.jar,
org.eclipse.ui.workbench.nl1_3.2.0.v200606220026.jar,
org.eclipse.ui.workbench.texteditor.nl1_3.2.0.v200606220026.jar,
org.eclipse.update.configurator.nl1_3.2.0.v200606220026.jar,
org.eclipse.update.core.nl1_3.2.0.v200606220026.jar,
org.eclipse.update.scheduler.nl1_3.2.0.v200606220026.jar,
org.eclipse.update.ui.nl1_3.2.0.v200606220026.jar,
com.ibm.icu_3.4.4.1.jar,
org.eclipse.core.commands_3.2.0.I20060605-1400.jar,
org.eclipse.core.contenttype_3.2.0.v20060603.jar,
org.eclipse.core.expressions_3.2.0.v20060605-1400.jar,
org.eclipse.core.filesystem.linux.x86_1.0.0.v20060603.jar,
org.eclipse.core.filesystem_1.0.0.v20060603.jar, org.eclipse.core.jobs_3.2.0.v20060603.jar,
org.eclipse.core.runtime.compatibility.auth_3.2.0.v20060601.jar,
org.eclipse.core.runtime_3.2.0.v20060603.jar, org.eclipse.equinox.common_3.2.0.v20060603.jar,
org.eclipse.equinox.preferences_3.2.0.v20060601.jar, org.eclipse.equinox.registry_3.2.0.v20060601.jar,
org.eclipse.help_3.2.0.v20060602.jar, org.eclipse.jface.text_3.2.0.v20060605-1400.jar,
org.eclipse.jface_3.2.0.I20060605-1400.jar, org.eclipse.osgi_3.2.0.v20060601.jar,
org.eclipse.swt.gtk.linux.x86_3.2.0.v3232m.jar, org.eclipse.swt_3.2.0.v3232o.jar,
org.eclipse.text_3.2.0.v20060605-1400.jar,
org.eclipse.ui.workbench.texteditor_3.2.0.v20060605-1400.jar,
org.eclipse.ui.workbench_3.2.0.I20060605-1400.jar, org.eclipse.ui_3.2.0.I20060605-1400.jar,
runtime_registry_compatibility.jar, eclipse.exe, eclipse.ini, and startup.jar
(collectively, "Eclipse Software").
All Eclipse Software is distributed under the terms and conditions of the Eclipse Foundation Software User Agreement (EFSUA) and/or terms and conditions of the Eclipse Public License Version 1.0 (EPL) or other license agreements, notices or terms and conditions referenced for the individual pieces of the Eclipse Software, including without limitation "Abouts", "Feature Licenses", and "Feature Update Licenses" as defined in the EFSUA .

A copy of the Eclipse Foundation Software User Agreement is found at
<install_dir>/SI/repository/rcp/rcpdependencies/windows/eclipse/notice.html,
<install_dir>/SI/repository/rcp/rcpdependencies/windows/eclipse/plugins/notice.html,
<install_dir>/SI/repository/rcp/rcpdependencies/gtk.linux_x86/eclipse/notice.html, and
<install_dir>/SI/repository/rcp/rcpdependencies/gtk.linux_x86/eclipse/plugins/notice.html.

A copy of the EPL is found at
<install_dir>/SI/repository/rcp/rcpdependencies/windows/eclipse/plugins/epl-v10.htm,
<install_dir>/SI/repository/rcp/rcpdependencies/windows/eclipse/epl-v10.htm,
<install_dir>/SI/repository/rcp/rcpdependencies/gtk.linux.x86/eclipse/plugins/epl-v10.html, and
<install_dir>/SI/repository/rcp/rcpdependencies/gtk.linux.x86/eclipse/epl-v10.html.

The reference to the license agreements, notices or terms and conditions governing each individual piece of the Eclipse Software is found in the directory files for the individual pieces of the Eclipse Software as described in the file identified as installdir/SCI_License.txt.

These licenses only apply to the Eclipse Software and do not apply to the Sterling Commerce Software, or any other Third Party Software.

The Language Pack (NL Pack) piece of the Eclipse Software, is distributed in object code form. Source code is available at http://archive.eclipse.org/eclipse/downloads/drops/L-3.2_Language_Packs-200607121700/index.php. In the event the source code is no longer available from the website referenced above, contact Sterling Commerce at 978-513-6000 and ask for the Release Manager. A copy of this license is located at <install_dir>/SI/repository/rcp/rcpdependencies/windows/eclipse/plugins/epl-v10.htm and

<install_dir>/SI/repository/rcp/rcpdependencies/gtk.linux.x86/eclipse/plugins/epl-v10.html.

The org.eclipse.core.runtime_3.2.0.v20060603.jar piece of the Eclipse Software was modified slightly in order to remove classes containing encryption items. The org.eclipse.core.runtime_3.2.0.v20060603.jar was modified to remove the Cipher, CipherInputStream and CipherOutputStream classes and rebuild the org.eclipse.core.runtime_3.2.0.v20060603.jar.

Ehcache Software

The Sterling Commerce Software is also distributed with or on the same storage media as the ehache v.1.5 software (Copyright © 2003-2008 Luck Consulting Pty. Ltd.) ("Ehache Software"). Ehcache Software is free software which is distributed under the terms of the Apache License Version 2.0. A copy of License Version 2.0 is found in <install>/jar/smcfs/8.5/ehcache-1.5.0.jar (./LICENSE.txt).

The Ehcache Software was not modified. Neither the Sterling Commerce Software, modifications, if any, to the Ehcache Software, nor other Third Party Code is a Derivative Work or a Contribution as defined in License Version 2.0. License Version 2.0 applies only to the Ehcache Software which is the subject of the specific directory file and does not apply to the Sterling Commerce Software or to any other Third Party Software. License Version 2.0 includes the following provision:

"Unless required by applicable law or agreed to in writing, Licensor provides the Work (and each Contributor provides its Contributions) on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied, including, without limitation, any warranties or conditions of TITLE, NON-INFRINGEMENT, MERCHANTABILITY, or FITNESS FOR A PARTICULAR PURPOSE. You are solely responsible for determining the appropriateness of using or redistributing the Work and assume any risks associated with Your exercise of permissions under this License."

EZMorph Software

The Sterling Commerce Software is also distributed with or on the same storage media as the EZMorph v. 1.0.4 software (Copyright © 2006-2008 Andres Almiray) ("EZMorph Software"). EZMorph Software is free software which is distributed under the terms of the Apache License Version 2.0. A copy of License Version 2.0 is found in <install>/jar/ezmorph/1_0_4/ezmorph-1.0.4.jar (./LICENSE.txt).

The EZMorph Software was not modified. Neither the Sterling Commerce Software, modifications, if any, to the EZMorph Software, nor other Third Party Code is a Derivative Work or a Contribution as defined in License Version 2.0. License Version 2.0 applies only to the EZMorph Software which is the subject of the specific directory file and does not apply to the Sterling Commerce Software or to any other Third Party Software. License Version 2.0 includes the following provision:

"Unless required by applicable law or agreed to in writing, Licensor provides the Work (and each Contributor provides its Contributions) on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied, including, without limitation, any warranties or conditions of TITLE, NON-INFRINGEMENT, MERCHANTABILITY, or FITNESS FOR A PARTICULAR PURPOSE. You are solely responsible for determining the appropriateness of using or redistributing the Work and assume any risks associated with Your exercise of permissions under this License."

Firebug Lite Software

The Sterling Commerce Software is distributed with or on the same storage media as the Firebug Lite Software which is free software distributed under the terms of the following license:

Copyright (c) 2008 Azer Koçulu http://azer.kodfabrik.com. All rights reserved.

ICE SOFTWARE

The Sterling Commerce Software is distributed on the same storage media as the ICE Software (Copyright © 1997 ICE Engineering, Inc./Timothy Gerard Endres.) ("ICE Software"). The ICE Software is independent from and not linked or compiled with the Sterling Commerce Software. The ICE Software is a free software product which can be distributed and/or modified under the terms of the GNU General Public License as published by the Free Software Foundation; either version 2 of the License or any later version.

A copy of the GNU General Public License is provided at installdir/jar/jniregistry/1_2/ICE_License.txt. This license only applies to the ICE Software and does not apply to the Sterling Commerce Software, or any other Third Party Software.

The ICE Software was modified slightly in order to fix a problem discovered by Sterling Commerce involving the RegistryKey class in the RegistryKey.java in the JNIRegistry.jar. The class was modified to comment out the finalize () method and rebuild of the JNIRegistry.jar file.

Source code for the bug fix completed by Sterling Commerce on January 8, 2003 is located at: installdir/jar/jniregistry/1_2/RegistryKey.java. Source code for all other components of the ICE Software is located at http://www.trustice.com/java/jnireg/index.shtml.

The ICE Software is distributed WITHOUT ANY WARRANTY; without even the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE.

JBOSS SOFTWARE

The Sterling Commerce Software is distributed on the same storage media as the JBoss Software (Copyright © 1999-2002 JBoss.org) ("JBoss Software"). The JBoss Software is independent from and not linked or compiled with the Sterling Commerce Software. The JBoss Software is a free software product which can be distributed and/or modified under the terms of the GNU Lesser General Public License as published by the Free Software Foundation; either version 2.1 of the License or any later version.

A copy of the GNU Lesser General Public License is provided at:
<install_dir>\jar\jboss\4_2_0\LICENSE.html

This license only applies to the JBoss Software and does not apply to the Sterling Commerce Software, or any other Third Party Software.

The JBoss Software is not distributed by Sterling Commerce in its entirety. Rather, the distribution is limited to the following jar files: el-api.jar, jasper-compiler-5.5.15.jar, jasper-el.jar, jasper.jar, jboss-common-client.jar, jboss-j2ee.jar, jboss-jmx.jar, jboss-jsr77-client.jar, jbossmq-client.jar, jnpserver.jar, jsp-api.jar, servlet-api.jar, tomcat-juli.jar.

The JBoss Software was modified slightly in order to allow the ClientSocketFactory to return a socket connected to a particular host in order to control the host interfaces, regardless of whether the ClientSocket Factory specified was custom or note. Changes were made to org.jnp..server.Main. Details concerning this change can be found at http://sourceforge.net/tracker/?func=detail&aid=1008902&group_id=22866&atid=376687.

Source code for the modifications completed by Sterling Commerce on August 13, 2004 is located at: http://sourceforge.net/tracker/?func=detail&aid=1008902&group_id=22866&atid=376687. Source code for all other components of the JBoss Software is located at http://www.jboss.org.

JGO SOFTWARE

The Sterling Commerce Software is distributed with, or on the same storage media, as certain redistributable portions of the JGo Software provided by Northwoods Software Corporation under a commercial license agreement (the "JGo Software"). The JGo Software is provided subject to the disclaimers set forth above and the following notice:

U.S. Government Restricted Rights

The JGo Software and documentation are provided with RESTRICTED RIGHTS. Use, duplication, or disclosure by the Government is subject to restrictions as set forth in subparagraph (C)(1)(ii) of the Rights in Technical Data and Computer Software clause at DFARS 252.227-7013 or subparagraphs (C)(1) and (2) of the Commercial Computer Software - Restricted Rights at 48 CFR 52.227-19, as applicable. Contractor / manufacturer of the JGo Software is Northwoods Software Corporation, 142 Main St., Nashua, NH 03060.

JSLib Software

The Sterling Commerce Software is distributed with or on the same storage media as the JSLib software product (Copyright (c) 2003-2009 Mozdev Group, Inc.) ("JSLib Software"). The JSLib Software is distributed under the terms of the MOZILLA PUBLIC LICENSE Version 1.1. A copy of this license is located at <install>\repository\eardata\platform_uifwk_ide\war\designer\MPL-1.1.txt. The JSLib Software code is distributed in source form and is located at http://jslib.mozdev.org/installation.html. Neither the Sterling Commerce Software nor any other Third-Party Code is a Modification or Contribution subject to the Mozilla Public License. Pursuant to the terms of the Mozilla Public License, the following notice applies only to the JSLib Software (and not to the Sterling Commerce Software or any other Third-Party Software):

"The contents of the file located at http://www.mozdev.org/source/browse/jslib/ are subject to the Mozilla Public License Version 1.1 (the "License"); you may not use this file except in compliance with the License. You may obtain a copy of the License at http://www.mozilla.org/MPL/MPL-1.1.html.

Software distributed under the License is distributed on an "AS IS" basis, WITHOUT WARRANTY OF ANY KIND, either express or implied. See the License for the specific language governing rights and limitations under the License.

The Original Code is Mozdev Group, Inc. code. The Initial Developer of the Original Code is Mozdev Group, Inc. Portions created by_Mozdev Group, Inc. are Copyright © 2003 Mozdev Group, Inc. All Rights Reserved. Original Author: Pete Collins <pete@mozdev.org>one Contributor(s):_____none listed________.

Alternatively, the contents of this file may be used under the terms of the ____ license (the "[___] License"), in which case the provisions of [___] License are applicable instead of those above. If you wish to allow use of your version of this file only under the terms of the [___] License and not allow others to use your version of this file under the MPL, indicate your decision by deleting the provisions above and replace them with the notice and other provisions required by the [___] License. If you do not delete the provisions above, a recipient may use your version of this file under either the MPL or the [___] License."

The preceding license only applies to the JSLib Software and does not apply to the Sterling Commerce Software, or any other Third-Party Software.

Json Software

The Sterling Commerce Software is also distributed with or on the same storage media as the Json 2.2.2 software (Copyright © 2006-2008 Json-lib) ("Json Software"). Json Software is free software which is distributed under the terms of the Apache License Version 2.0. A copy of License Version 2.0 is found in <install>/jar/jsonlib/2_2_2/json-lib-2.2.2-jdk13.jar.

This product includes software developed by Douglas Crockford (http://www.crockford.com).

The Json Software was not modified. Neither the Sterling Commerce Software, modifications, if any, to the Json Software, nor other Third Party Code is a Derivative Work or a Contribution as defined in License Version 2.0. License Version 2.0 applies only to the Json Software which is the subject of the specific directory file and does not apply to the Sterling Commerce Software or to any other Third Party Software. License Version 2.0 includes the following provision:

"Unless required by applicable law or agreed to in writing, Licensor provides the Work (and each Contributor provides its Contributions) on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied, including, without limitation, any warranties or conditions of TITLE, NON-INFRINGEMENT, MERCHANTABILITY, or FITNESS FOR A PARTICULAR PURPOSE. You are solely responsible for determining the appropriateness of using or redistributing the Work and assume any risks associated with Your exercise of permissions under this License."

Purple Technology

The Sterling Commerce Software is distributed with or on the same storage media as the Purple Technology Software (Copyright (c) 1995-1999 Purple Technology, Inc.) ("Purple Technology Software"), which is subject to the following license:

Copyright (c) 1995-1999 Purple Technology, Inc. All rights reserved.

PLAIN LANGUAGE LICENSE: Do whatever you like with this code, free of charge, just give credit where credit is due. If you improve it, please send your improvements to alex@purpletech.com. Check http://www.purpletech.com/code/ for the latest version and news.

LEGAL LANGUAGE LICENSE: Redistribution and use in source and binary forms, with or without modification, are permitted provided that the following conditions are met:

1. Redistributions of source code must retain the above copyright notice, this list of conditions and the following disclaimer.

2. Redistributions in binary form must reproduce the above copyright notice, this list of conditions and the following disclaimer in the documentation and/or other materials provided with the distribution.

3. The names of the authors and the names "Purple Technology," "Purple Server" and "Purple Chat" must not be used to endorse or promote products derived from this software without prior written permission. For written permission, please contact server@purpletech.com.

THIS SOFTWARE IS PROVIDED BY THE AUTHORS AND PURPLE TECHNOLOGY "AS IS'' AND ANY EXPRESSED OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL THE AUTHORS OR PURPLE TECHNOLOGY BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

The preceding license only applies to the Purple Technology Software and does not apply to the Sterling Commerce Software, or any other Third Party Software.

Rico Software

The Sterling Commerce Software is also distributed with or on the same storage media as the Rico.js software (Copyright © 2005 Sabre Airline Solutions) ("Rico Software"). Rico Software is free software which is distributed under the terms of the Apache License Version 2.0. A copy of License Version 2.0 is found in <install>/repository/eardata/platform/war/ajax/scripts/Rico_License.txt.

The Rico Software was not modified. Neither the Sterling Commerce Software, modifications, if any, to the Rico Software, nor other Third-Party Code is a Derivative Work or a Contribution as defined in License Version 2.0. License Version 2.0 applies only to the Rico Software which is the subject of the specific directory file and does not apply to the Sterling Commerce Software or to any other Third-Party Software. License Version 2.0 includes the following provision:

"Unless required by applicable law or agreed to in writing, Licensor provides the Work (and each Contributor provides its Contributions) on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied, including, without limitation, any warranties or conditions of TITLE, NON-INFRINGEMENT, MERCHANTABILITY, or FITNESS FOR A PARTICULAR PURPOSE. You are solely responsible for determining the appropriateness of using or redistributing the Work and assume any risks associated with Your exercise of permissions under this License."

Rhino Software

The Sterling Commerce Software is distributed with or on the same storage media as the Rhino js.jar (Copyright (c) 1998-2009 Mozilla.org.) ("Rhino Software"). A majority of the source code for the Rhino Software is dual licensed under the terms of the MOZILLA PUBLIC LICENSE Version 1.1. or the GPL v. 2.0. Additionally, some files (at a minimum the contents of toolsrc/org/Mozilla/javascript/toolsdebugger/treetable) are available under another license as set forth in the directory file for the Rhino Software.

Sterling Commerce's use and distribution of the Rhino Software is under the Mozilla Public License. A copy of this license is located at <install>/3rdParty/rico license.doc. The Rhino Software code is distributed in source form and is located at http://mxr.mozilla.org/mozilla/source/js/rhino/src/. Neither the Sterling Commerce Software nor any other Third-Party Code is a Modification or Contribution subject to the Mozilla Public License. Pursuant to the terms of the Mozilla Public License, the following notice applies only to the Rhino Software (and not to the Sterling Commerce Software or any other Third-Party Software):

"The contents of the file located at <install>/jar/rhino/1_7R1/js.jar are subject to the Mozilla Public License Version 1.1 (the "License"); you may not use this file except in compliance with the License. You may obtain a copy of the License at http://www.mozilla.org/MPL/.

Software distributed under the License is distributed on an "AS IS" basis, WITHOUT WARRANTY OF ANY KIND, either express or implied. See the License for the specific language governing rights and limitations under the License.

The Original Code is Rhino code, released May 6, 1999. The Initial Developer is Netscape Communications Corporation. Portions created by the Initial Developer are Copyright © 1997-1999. All Rights Reserved. Contributor(s):_____none listed.

The preceding license only applies to the Rico Software and does not apply to the Sterling Commerce Software, or any other Third-Party Software.

Sun Microsystems

The Sterling Commerce Software is distributed with or on the same storage media

as the following software products (or components thereof): Sun JMX, and Sun JavaMail (collectively, "Sun Software"). Sun Software is free software which is distributed under the terms of the licenses issued by Sun which are included in the directory files located at:

SUN COMM JAR - <install>/Applications/Foundation/lib

SUN ACTIVATION JAR - <install>/ Applications/Foundation/lib

SUN JavaMail - <install>/jar/javamail/1_4/LICENSE.txt

The Sterling Commerce Software is also distributed with or on the same storage media as the Web-app_2_3.dtd software (Copyright © 2007 Sun Microsystems, Inc.) ("Web-App Software"). Web-App Software is free software which is distributed under the terms of the Common Development and Distribution License ("CDDL"). A copy of the CDDL is found in http://kenai.com/projects/javamail/sources/mercurial/show.

The source code for the Web-App Software may be found at: <install>/3rdParty/sun/javamail-1.3.2/docs/JavaMail-1.2.pdf

Such licenses only apply to the Sun product which is the subject of such directory and does not apply to the Sterling Commerce Software or to any other Third Party Software.

The Sterling Commerce Software is also distributed with or on the same storage media as the Sun Microsystems, Inc. Java (TM) look and feel Graphics Repository ("Sun Graphics Artwork"), subject to the following terms and conditions:

Copyright 2000 by Sun Microsystems, Inc. All Rights Reserved.

Sun grants you ("Licensee") a non-exclusive, royalty free, license to use, and redistribute this software graphics artwork, as individual graphics or as a collection, as part of software code or programs that you develop, provided that i) this copyright notice and license accompany the software graphics artwork; and ii) you do not utilize the software graphics artwork in a manner which is disparaging to Sun. Unless enforcement is prohibited by applicable law, you may not modify the graphics, and must use them true to color and unmodified in every way.

This software graphics artwork is provided "AS IS," without a warranty of any kind. ALL EXPRESS OR IMPLIED CONDITIONS, REPRESENTATIONS AND WARRANTIES, INCLUDING ANY IMPLIED WARRANTY OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE OR NON-INFRINGEMENT, ARE HEREBY EXCLUDED. SUN AND ITS LICENSORS SHALL NOT BE LIABLE FOR ANY DAMAGES SUFFERED BY LICENSEE AS A RESULT OF USING, MODIFYING OR DISTRIBUTING THE SOFTWARE GRAPHICS ARTWORK.

IN NO EVENT WILL SUN OR ITS LICENSORS BE LIABLE FOR ANY LOST REVENUE, PROFIT OR DATA, OR FOR DIRECT, INDIRECT, SPECIAL, CONSEQUENTIAL, INCIDENTAL OR PUNITIVE DAMAGES, HOWEVER CAUSED AND REGARDLESS OF THE THEORY OF LIABILITY, ARISING OUT OF THE USE OF OR INABILITY TO USE SOFTWARE GRAPHICS ARTWORK, EVEN IF SUN HAS BEEN ADVISED OF THE POSSIBILITY OF SUCH DAMAGES.

If any of the above provisions are held to be in violation of applicable law, void, or unenforceable in any jurisdiction, then such provisions are waived to the extent necessary for this Disclaimer to be otherwise enforceable in such jurisdiction.

The preceding license only applies to the Sun Graphics Artwork and does not apply to the Sterling Commerce Software, or any other Third Party Software.

WARRANTY DISCLAIMER

This documentation and the Sterling Commerce Software which it describes are licensed either "AS IS" or with a limited warranty, as set forth in the Sterling Commerce license agreement. Other than any limited warranties provided, NO OTHER WARRANTY IS EXPRESSED AND NONE SHALL BE IMPLIED, INCLUDING THE WARRANTIES OF MERCHANTABILITY AND FITNESS FOR USE OR FOR A PARTICULAR PURPOSE. The applicable Sterling Commerce entity reserves the right to revise this publication from time to time and to make changes in the content hereof without the obligation to notify any person or entity of such revisions or changes.

The Third Party Software is provided "AS IS" WITHOUT ANY WARRANTY AND ANY EXPRESSED OR IMPLIED WARRANTIES, INCLUDING BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY, AND FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED. FURTHER, IF YOU ARE LOCATED OR ACCESSING THIS SOFTWARE IN THE UNITED STATES, ANY EXPRESS OR IMPLIED WARRANTY REGARDING TITLE OR NON-INFRINGEMENT ARE DISCLAIMED.

Without limiting the foregoing, the ICE Software and JBoss Software are distributed WITHOUT ANY WARRANTY; without even the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE.

Contents

1

Checklist for Customization Projects

This chapter provides a high-level checklist for the tasks involved in customizing or extending Selling and Fulfillment Foundation.

1.1 Customization Projects

Projects to customize or extend Selling and Fulfillment Foundation vary with the type of changes that are needed. However, most projects involve an interconnected series of changes that are best carried out in a particular order. The checklist identifies the most common order of customization tasks and indicates which guide in the documentation set provides details about each stage.

1. Prepare your development environment

   Set up a development environment that mirrors your production environment, including whether you deploy Selling and Fulfillment Foundation on a WebLogic, WebSphere, or JBoss application server. Doing so ensure that you can test your extensions in a real-time environment.

   You install and deploy Selling and Fulfillment Foundation in your development environment following the same steps that you used to install and deploy Selling and Fulfillment Foundation in your production environment. Refer to Selling and Fulfillment Foundation system requirements and installation documentation for details.

   An option is to customize Selling and Fulfillment Foundation with Microsoft COM+. Using COM+ provides you with advantages such as increased security, better performance, increased manageability of server applications, and support for clients of mixed environments. If this is your choice, see the *Selling and Fulfillment Foundation: Customization Basics Guide* about additional installation instructions.

2. Plan your customizations

   Are you adding a new menu entry, customizing the Sign In screen and logo, creating new themes, customizing views and wizards, or adding new screens? Each type of customization varies in scope and complexity. For background, see the *Selling and Fulfillment Foundation: Customization Basics Guide*, which summarizes the types of changes that you can make.

   Important guidelines about file names, keywords, and other conventions are found in the *Selling and Fulfillment Foundation: Customization Basics Guide*.

3. Extend the Database

   For many customization projects, the first task is to extend the database so that it supports the other UI or API changes that you make later. For instructions, see the *Selling and Fulfillment Foundation: Extending the Database Guide* which include information about the following topics:

   - Important guidelines about what you can and cannot change in the database.
   - Information about modifying APIs. If you modify database tables so that any APIs are impacted, you must extend the templates of those APIs or you cannot store or retrieve data from the database. This step is required if table modifications impact an API.
   - How to generate audit references so that you improve record management by tracking records at the entity level. This step is optional.

4. Make other changes to APIs

   Selling and Fulfillment Foundation can call or invoke standard APIs or custom APIs. For background about APIs and the services architecture in Selling and Fulfillment Foundation, including service types, behavior, and security, see the *Selling and Fulfillment Foundation: Customizing APIs Guide*. This guide includes information about the following types of changes:

- How to invoke standard APIs for displaying data in the UI and also how to .save the changes made to the UI in the database.
- Invoke customized APIs for executing your custom logic in the extended service definitions and pipeline configurations.
- APIs use input and output XML to store and retrieve data from the database. If you don't extend these API input and output XML files, you may not get the results you want in the UI when your business logic is executing.
- Every API input and output XML file has a DTD and XSD associated to it. Whenever you modify input and output XML, you must generate the corresponding DTD and XSD to ensure data integrity. If you don't generate the DTD and XSD for extended Application XMLs, you may get inconsistent data.

5. Customize the UI

Sterling Commerce applications support several UI frameworks. Depending on your application and the customizations you want to make, you may work in only one or in several of these frameworks. Each framework has its own process for customizing components like menu items, logos, themes, and etc. Depending on the framework you want, consult one of the following guides:

- *Selling and Fulfillment Foundation: Customizing Console JSP Interface for End User Guide*
- *Selling and Fulfillment Foundation: Customizing the Swing Interface Guide*
- *Selling and Fulfillment Foundation: Customizing User Interfaces for Mobile Devices Guide*
- *Selling and Fulfillment Foundation: Customizing the RCP Interface Guide* and *Selling and Fulfillment Foundation: Using the Sterling RCP Extensibility Tool Guide*
- *Customizing the Web UI Framework Guide*

6. Extend Transactions

You can extend the standard Selling and Fulfillment Foundation to enhance the functionality of your implementation of Selling and Fulfillment Foundation and to integrate with external systems. For background about transaction types, security, dynamic variables, and extending the

Condition Builder, see the *Selling and Fulfillment Foundation: Extending Transactions Guide Selling and Fulfillment Foundation: Extending the Condition Builder Guide* . These guides includes information about the following types of changes:

- How to extend Selling and Fulfillment Foundation Condition Builder to define complex and dynamic conditions for executing your custom business logic and using a static set of attributes.
- How to define variables to dynamically configure properties belonging to actions, agents, and services configurations.
- How to set up transactional data security for controlling who has access to what data, how much they can see, and what they can do with it.
- How to create custom time-triggered transactions. You can invoke and schedule these custom time-triggered transactions in much the same manner as you invoke and schedule Selling and Fulfillment Foundation standard time-triggered transactions. Finally, you can coordinate your custom, time-triggered transactions with external transactions and run them either by raising an event, calling a user exit, or invoking a custom API or service.

7. Build and deploy your customizations or extensions

After performing the customizations that you want, you must build and deploy your customizations or extensions. First, build and deploy these customizations or extensions in the test environment for verification. When you are ready, repeat the same process to build and deploy your customizations and extensions in the production environment. For instructions, see the *Selling and Fulfillment Foundation: Customization Basics Guide* which includes information about the following topics:

- How to build and deploy standard resources, database, and other extensions (such as templates, user exits, java interfaces).
- How to build and deploy Enterprise-Level extensions.

2

Basics of Using the Rich Client Platform Extensibility Tool

2.1 Getting Started with the Rich Client Platform Extensibility Tool

The Rich Client Platform Extensibility Tool allows you to extend the Rich Client Platform UI by adding new controls, modifying existing controls, and so forth. The tool facilitates the adding of UI controls such as labels, text boxes, combo boxes, list boxes, and so forth. The Rich Client Platform Extensibility Tool provides the ability to change the properties of the existing Rich Client Platform-provided fields. You can also add new fields and specify the layout, bindings, and theme properties for these fields. The Rich Client Platform Extensibility Tool also allows you to synchronize the differences in theme entries, bundle entries, and templates.

> **Note:** Some screens in the Rich Client Platform application cannot be extended using the Rich Client Platform Extensibility Tool. When you perform any operation on such screens using the Rich Client Platform Extensibility Tool, the following message displays:
>
> *Cannot extend an Inextensible Screen*

2.1.1 Starting the Rich Client Platform Extensibility Tool

Before you start extending the screens, you need to start the Rich Client Platform Extensibility Tool.

To extend a screen:

1. From the Rich Client Platform application’s menu bar, select File > Extend. The Rich Client Platform Extensibility Tool opens.
2. Click  to load the extension file, if applicable.
3. Click  to start extending the screen.

2.1.2 Loading the Rich Client Platform Extension File

Before you extend or modify the existing forms using the Rich Client Platform Extensibility Tool, you must load `<Plug-in id>_extn.yuix` extension file in the Rich Client Platform Extensibility Tool. To open the Rich Client Platform Extensibility Tool, from the menu bar, select File > Extend. All registered extension files automatically gets loaded in the tool, and displays in the bottom panel. If the extension files are unregistered, you must register them and then load the files prior to making changes to the form.

To load the extension files:

1. From the menu bar, select File > Extend. The Rich Client Platform Extensibility Tool opens.
2. Click . The Open dialog displays.
3. Select the `<Plug-in id>_extn.yuix` extension file.

If you do not load `<Plug-in id>_extn.yuix` extension file before extending or modifying forms, the Load/Create Extension File window displays. You can either load an existing extension file or create and load the new extension file.

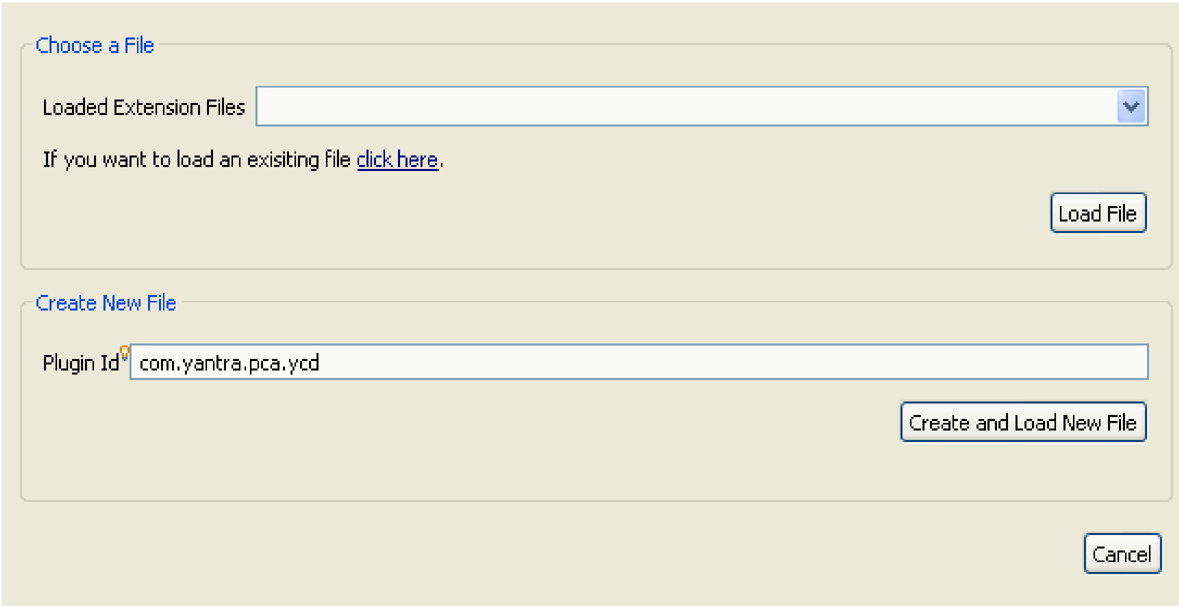


Table 2–1 Load/Create Extension File, Choose a File

Field	Description
Choose a File	
Loaded Extension Files	Select the extension file from the drop-down list, if applicable. If an extension file is not loaded in the tool, this combo box gets disabled.
click here	Click this hyperlink. The Choose File dialog displays. Select an extension file in which a new form element is automatically created. Click open. The extension file that you selected automatically gets loaded in the tool.
Load File	Click the Load File button to load the extension file that you selected in the Loaded Extension Files combo box.
Create New File	

Table 2–1 Load/Create Extension File, Choose a File

Field	Description
Plugin Id	Enter the plugin identifier of the plug-in that registers this new extension file. Press Ctrl+Space to select the plug-in Id from the list of available plug-ins.
Create and Load New File	When you click this button, the Save File dialog displays. If you are creating a new extension file, enter the extension file name and click Save. The new extension file gets created and loaded automatically. Otherwise, select an existing extension file in which a new form element is automatically created. Click Save. The extension file gets automatically loaded in the tool.

2.2 Viewing Screen Information

You can view the following information for a screen:

- The identifier of the editor contained by the screen.
- The identifier of the form is used to identify the screen.
- The identifiers of the task associated with the an editor.
- The <Plug-in_id>_extn.yuix extensions file associated to a particular screen.
- List of all the commands used by the screen to call APIs or services to get the required data for the screen.
- List of all the namespaces defined for the screen.

To view screen information:

1. Start the Rich Client Platform Extensibility Tool.
2. Click Screen Info. The Screen Information window displays.

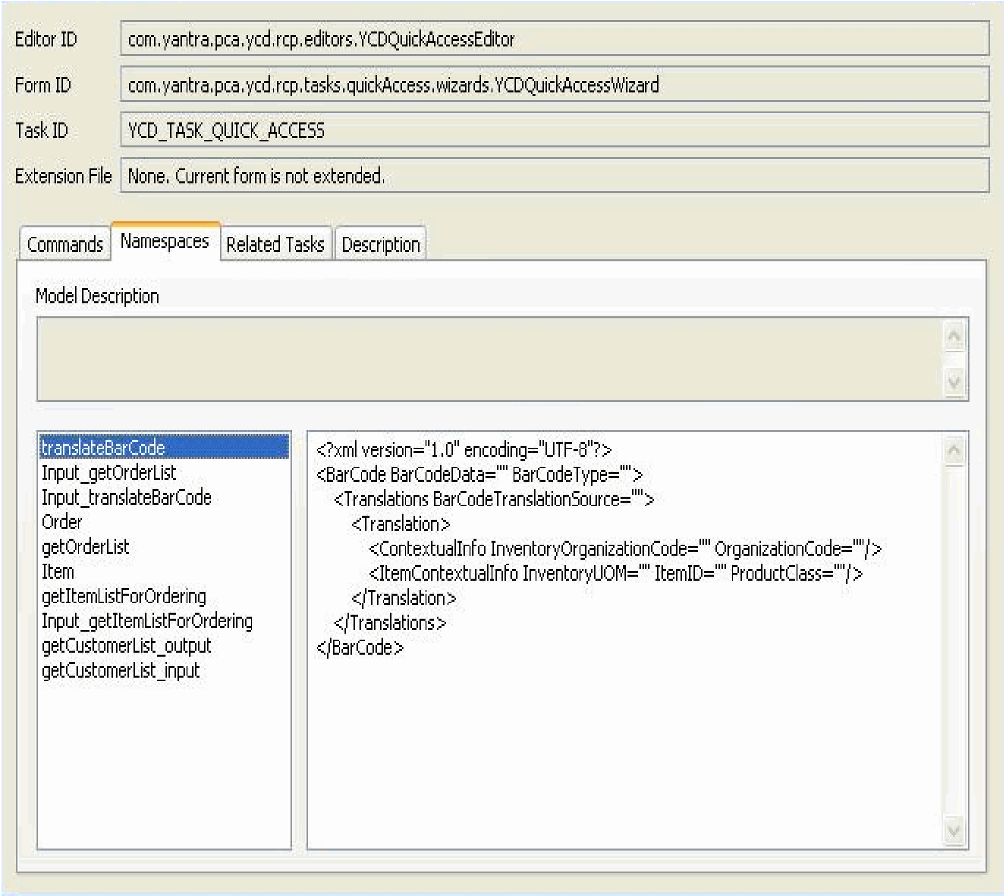


Table 2–2 Screen Information

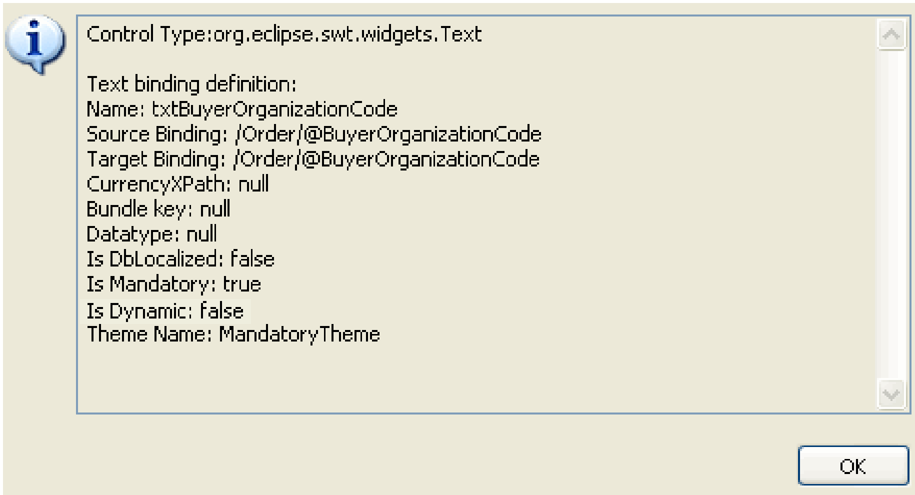
Field	Description
Editor ID	The identifier of the editor contained by the screen.
Form ID	The identifier of the form is used to identify the screen.
Task ID	The identifiers of the task associated with the editor.
Extension File	The <Plug-in_id>_extn.yuix extensions file associated with the screen.

3. Select the commands tab to view all the commands that are being used by the screen.
4. Select the Namespaces tab to view the list of namespaces that are used by the screen. You can also view the XML model of the output template associated with the a namespace in the right hand side panel. When you select a particular namespace in the left hand side panel, the description of that particular namespace is shown in the Model Description field.
5. Select the Related Tasks tab to view the list of category id's in which this current active task is interested in. You can also view the list of group id's along with their group sequence number associated with the related tasks of the current active task.
6. Select the Description tab to view the description of the wizard and wizard page. The Wizard Description field displays the description of the wizard and Current Page Description field displays the description of the current wizard page of the wizard.

2.3 Viewing Control Information

To view screen information:

1. Start the Rich Client Platform Extensibility Tool.
2. Click i Control Info. The Control Information window displays. Figure displays the sample control info screen for a text box control.



The system provides the following information for various controls:

- Control Type—The type of the control. For example, composite, group, button, and so forth.
- Name—The name of the control if applicable. For example, rdOpen, cmbPrice.
- Control Binding Definition—The information about the various bindings defined for a control. For example, source binding, target binding, checked binding.
- Theme Information - The theme applied to a control is displayed. For example, Theme Name: Mandatory Theme.

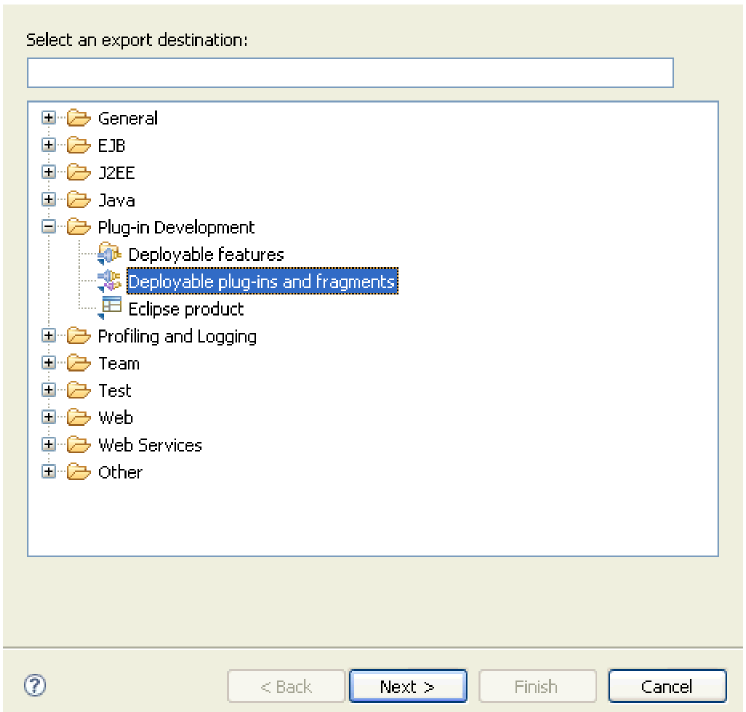
  For more information about binding definitions for each control, see the *Selling and Fulfillment Foundation: Javadocs*.

2.4 Building and Deploying Rich Client Platform Extensions

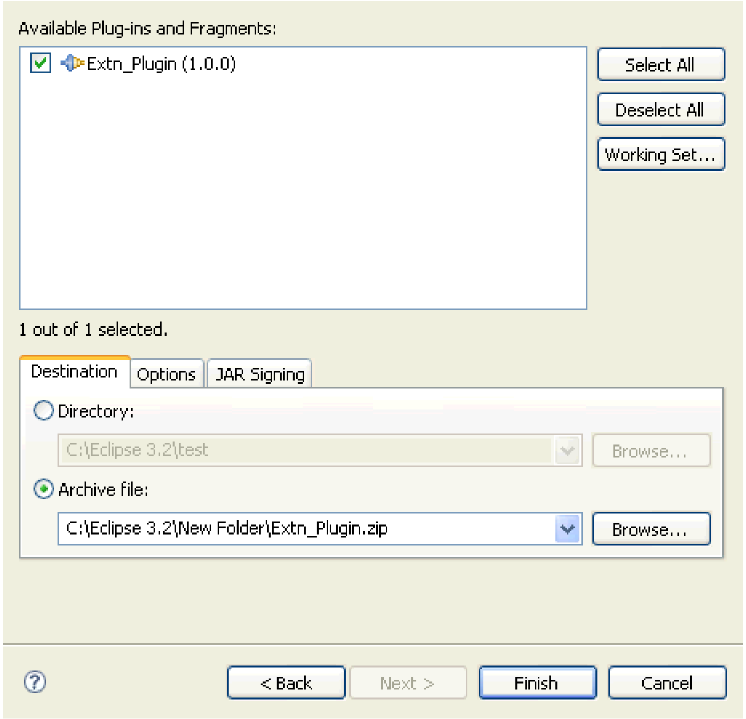
2.4.1 Building Rich Client Platform Extensions

Building the Rich Client Platform extensions is as follows:

1. Start the Eclipse SDK.
2. From the menu bar, select Window > Show View > Navigator. The plug-in project is displayed in the Navigator view.
3. Right-click on the plug-in project that you want to build and deploy.
4. Select Export... from the pop-up menu. The Export window displays.



5. From the list of export destinations, under Plug-in Deployment, select Deployable plug-ins and fragments.
6. Click Next.



7. In the Destination tab, Choose Archive file:.
8. Click Browse and browse to the folder where you want to store the exported plug-in zip file.
9. In the Options tab, make sure that the Package plug-ins as individual JAR archives box is checked.
10. Click Finish. The plug-in jar is generated and stored in the plugins folder in the zip file as specified in Step 8.

2.4.2 Deploying Rich Client Platform Extensions

After you build the Rich Client Platform extensions plugin jar, you must deploy this plug-in.

To deploy the Rich Client Platform extensions:

Copy the plugin jar that you built to the plugins directory of the <RCP_EXTN_FOLDER> folder and follow the steps as described in the "Deploying and Updating Rich Client Platform Application" chapter of the *Selling and Fulfillment Foundation: Installation Guide*.

3

Adding and Moving Controls and Table Columns in the Rich Client Platform UI

3.1 Adding a Label

To add a label:

1. Start the Rich Client Platform Extensibility Tool.
2. Click 

. To add a label, do one of the following:
   - Select the control where you want to add the label and click once.
   - Select the composite or group where you want to add the label and click once.

   The Add Label window displays.

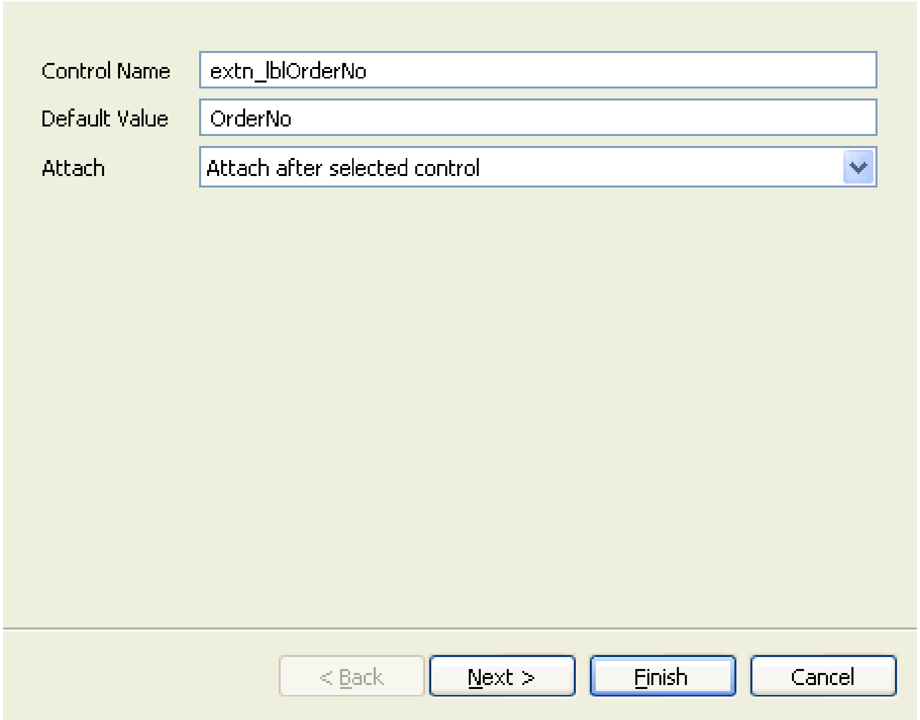


Table 3–1 Add Label

Field	Description
Control Name	Mandatory field. Enter a unique control name for the new label. Every field on the form must have a unique logical name for reference. (Optional) Prefix the control name with "extn_". If you do not specify this, the system automatically adds "extn_" to the control name.

Table 3–1 Add Label

Field	Description
Default Value	Mandatory field. Enter the default value to display on the screen.
Attach	Select the appropriate value from the drop-down list. Valid values are "Attach after selected control", "Attach before selected control", and "Attach within selected control". **Note:** If you are adding a label to an empty composite or group, by default, the "Attach within selected control" value displays.

3. Click Finish to add the new label with the default layout data, theme, and null binding attributes.
4. Click Next to specify the data layout options.

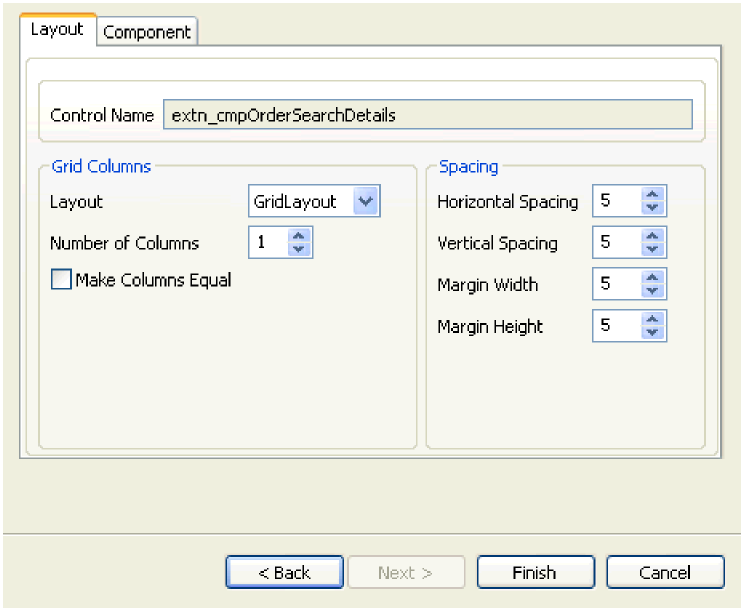


Table 3–2   Add Label, Layout Setup Page

<table>
<tr><th>Field</th><th>Description</th></tr>
<tr><td colspan="2">Layout Tab

This tab is used to set the layout properties for the control you want to add. Based on the control, you can set the layout properties for the following:
Parent Composite—If you want to add a control that is not a composite or group, in the layout tab you can set the layout properties of the parent control (composite or group) to which you want to add the new control. You can set the position and size of the control in the parent control. The positioning and sizing of a field depends on the layout of the parent control.
Composite or Group—If you want to add a composite or group, in the layout tab you can set the layout properties of the composite or group itself.
Note: If you add a control that is not a composite or group and the control name for its parent control (composite or group) is not set, the layout tab is disabled.</td></tr>
<tr><td>Control Name</td><td>Disabled field.

If you are adding a control which is not a composite or group, this field displays the name of the parent control (composite or group) to which you want to add the new control.

If you add a composite or group, this field displays the name of the composite or group itself.</td></tr>
<tr><td colspan="2">Grid Columns</td></tr>
<tr><td>Layout</td><td>Select the layout for the control. The valid value is "GridLayout".</td></tr>
<tr><td>Number of Columns</td><td>Enter a number or click to increase or decrease the number of columns for the layout.

Note: You can add one or more controls to the same row by changing the number of columns in the layout.</td></tr>
<tr><td>Make Columns Equal</td><td>Check this box if you want all columns in this layout to be of equal width.</td></tr>
<tr><td colspan="2">Spacing</td></tr>
<tr><td>Horizontal Spacing</td><td>If you increase or decrease the horizontal space, the horizontal space between the two neighboring columns also increases or decreases.</td></tr>
</table>

Table 3–2 Add Label, Layout Setup Page

Field	Description
Vertical Spacing	If you increase or decrease the vertical space, the vertical space between the two neighboring columns also increases or decreases.
Margin Width	If you increase or decrease the margin width, the horizontal margin along the left and right edges of the layout also increases or decreases.
Margin Height	If you increase or decrease the margin height, the vertical margin along the top and bottom edges of the layout also increases or decreases.
Component Tab Set the properties of the layout data of the control.	
Pick Layout Like	To apply the same layout as another field on the form, select the field from the drop-down list and view its layout. The drop-down list contains the control names that are siblings of the selected control.
Pick This	When you click this button all other fields are automatically populated based on the layout of the control you selected in the Pick Layout Like option.
Height	Enter any value between -1 and 9999 for the control.
Width	Enter any value between -1 and 9999 for the control.
Horizontal Span	If you increase the horizontal span by specifying any value between 1 and 25, the new control spans to the right of its current position.
Vertical Span	If you increase the vertical span by specifying any value between 1 and 25, the new control spans to the cell below its current position.
Grab Horizontal Space	Check this box if you want the new control to grab the extra horizontal space.
Grab Vertical Space	Check this box if you want the new control to grab the extra vertical space.

Table 3–2 Add Label, Layout Setup Page

Field	**Description**
Horizontal Alignment	Select the appropriate horizontal alignment type for the new control. Valid values are 'BEGINNING', 'CENTER', 'END', or 'FILL'. • BEGINNING aligns the new control with the left side of the horizontal space. • CENTER centers the new control within the horizontal space. • END aligns the new control with the right side of the horizontal space. • FILL fills the excess horizontal space.
Vertical Alignment	Select the appropriate vertical alignment type for the new control. Valid values are 'BEGINNING', 'CENTER', 'END', or 'FILL'. • BEGINNING aligns the new control with the top of the vertical space. • CENTER centers the new control within the vertical space. • END aligns the new control with the bottom of the vertical space. • FILL fills the excess vertical space.
Horizontal Indent	Set the horizontal indentation that you want to have on the left side of the control.

5. Click Finish to add the new label with the default theme and null binding attributes.
6. Click Next to specify the binding options.

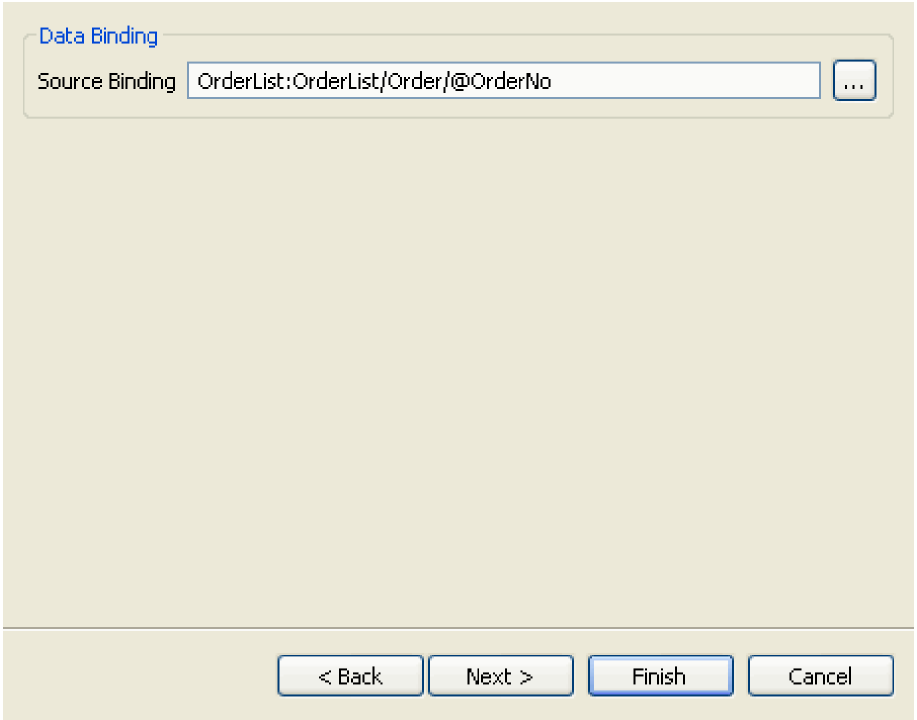


Table 3–3 Add Label, Binding Options Page

Field	Description
Source Binding	Enter the XML path to populate the new field, if applicable. Click ... to view the available XML paths. The Source Bindings Tree pop-up window displays. Select the appropriate XML path from the tree. You can also press Ctrl+Space and select the appropriate XML path from the drop-down list.

7. Click Finish to add the new label with the default theme binding.
8. Click Next to specify the theme binding.

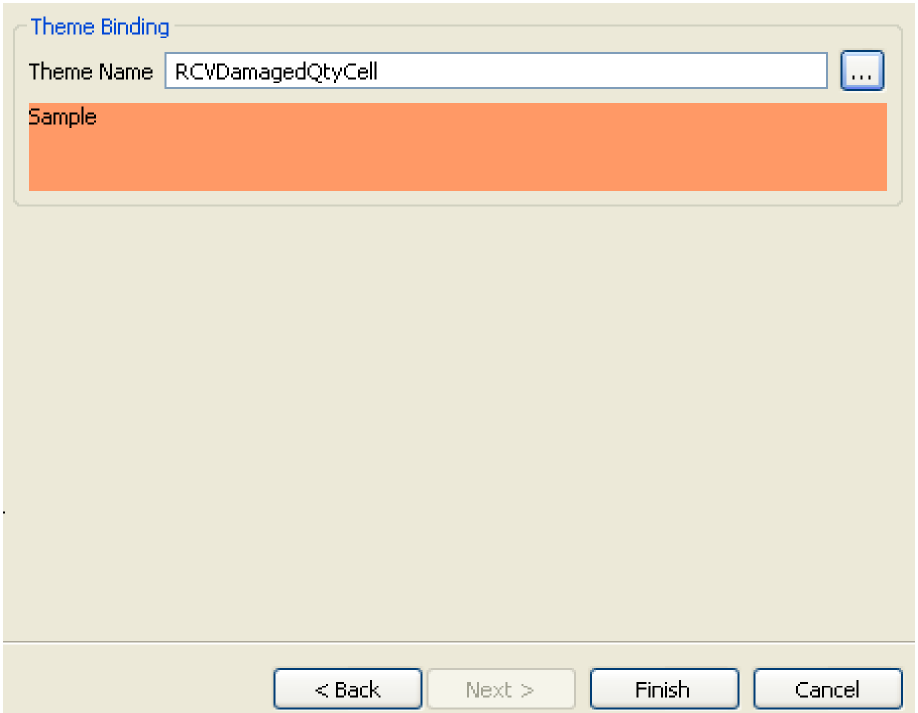


Table 3–4 Add Label, Theme Binding Page

Field	Description
Theme Name	Enter the theme name you want to apply for the new field, if applicable. Click  to view the available themes. The Themes Tree pop-up window is displayed, which lists all the available themes and also provides a preview of each theme. Select the appropriate theme name you want to apply. You can also press Ctrl+Space and select the appropriate theme name you want to apply from the drop-down list. The current theme applied to the selected control is displayed below the theme name. If an image is specified as the theme, then the image is displayed.

9. Click Finish.

10. Click to save the changes made to the extension file.

11. If you have specified the default value, source binding, or theme entry for the label, you must synchronize the resource files. For more information about synchronizing resource files, see Chapter 6, "Synchronizing New and Existing UI Resources".

3.2 Adding a Button

To add a button:

1. Start the Rich Client Platform Extensibility Tool.

2. Click Button .

3. Perform any of the following tasks to add a button:

   - Select the control where you want to add the button and click once.

   - Select the composite or group where you want to add the button and click once.

   The Add Button window displays.

> **Note:** You can place the button either in the row layout container or grid layout container. However, the row layout container does not have the layout data. Therefore, if you place the button in the row layout container, the Layout Setup Page is disabled.

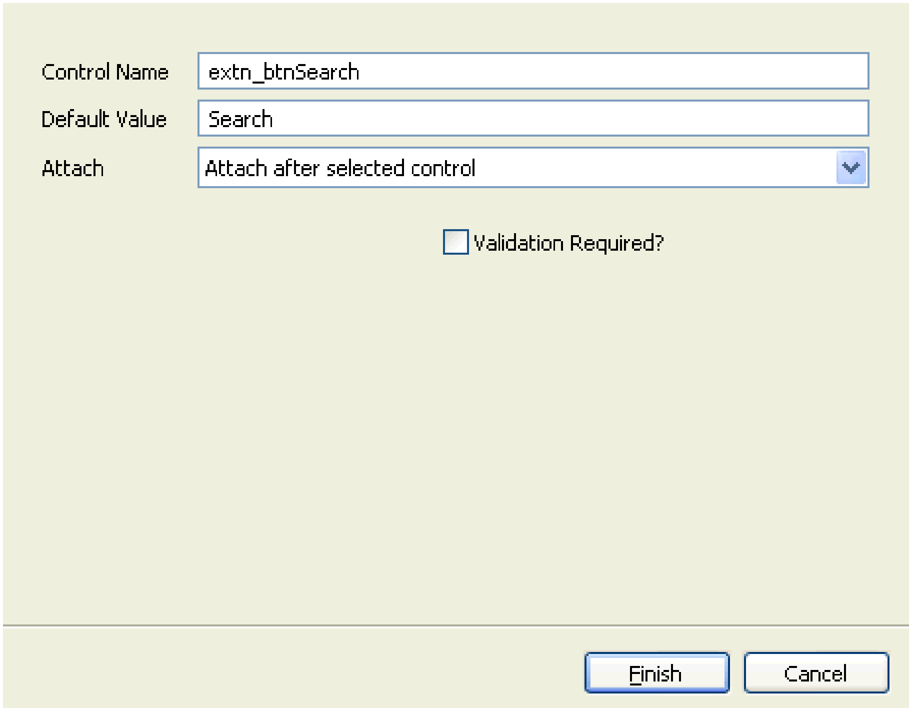


Table 3–5 Add Button

Field	Description
Control Name	Mandatory field. Enter a unique control name for the new control. Every field on the form must have a unique logical name for reference. (Optional) Prefix the control name with "extn_". If you do not specify this, the system automatically adds "extn_" to the control name.
Default Value	Mandatory field. Enter the default value to display on the screen.

Table 3–5 Add Button

Field	Description
Attach	Select the appropriate value from the drop-down list. Valid values are "Attach after selected control", "Attach before selected control", and "Attach within selected control". **Note:** If you are adding the new field to an empty composite or group, by default, the "Attach within selected control" value displays.
Validation Required?	Check this box to validate a new field, if applicable. The Rich Client Platform invokes the validation method on the extended behavior class of this form when the field loses focus.

4. Click Finish to add the new button with the default layout data, theme, and empty hot key binding.
5. Click Next to specify the layout data options. For field value descriptions, see Table 3–2.
6. Click Finish to add the new button with the default theme and empty hot key binding.
7. Click Next to specify the theme binding. For field value descriptions, see Table 3–4.
8. Click Next to specify the hot key binding.

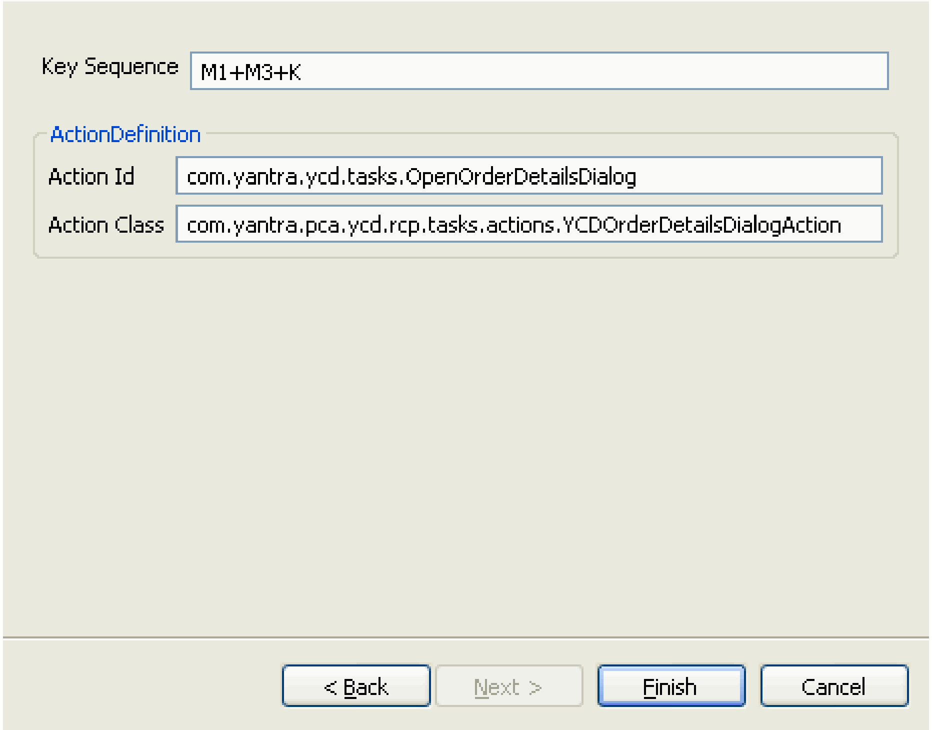


Table 3–6 Add Button, Hot key Binding

Field	Description
Key Sequence	Enter a valid key sequence of the hot key for the control. For example, F7, M1+K, and so forth. • Use M1 to specify the Ctrl key. • Use M2 to specify the Shift key. • Use M3 to specify the Alt key. To specify a combination of keys use the "+" operator. For example, to specify the hot key for a control as "Ctrl+Alt+K, enter the key sequence as "M1+M3+K".
Action Definition	

Table 3–6 Add Button, Hot key Binding

Field	Description
Action Id	Enter the identifier of the action you want to invoke upon pressing the hot key. **Note:** The action identifier specified in this field must not exist in the `plugin.xml` file of your Rich Client Platform extension plug-in.
Action Class	Enter the fully qualified java classpath of the action class you want to invoke upon pressing the hot key.

> **Note:** If you enter and save the hot key binding, you cannot modify it.

9. Click Finish.

   If you have specified the hot key binding, the Restart Application pop-up window displays and prompts you to save the extension files and restart the application to view the changes.

> **Note:** Whenever you specify the hot key binding for a control, you must save the extension file and restart the application to view the changes made to the hot key binding. Ensure that you clear the configuration data before starting the application.

10. Click  to save the changes made to the extension file.
11. If you have specified the default value for the button, you must synchronize the resource files. For more information about synchronizing the resource files, see Chapter 6, "Synchronizing New and Existing UI Resources".
12. (Optional) To implement the logic on the button click event:

> **Note:** When adding the new button, make sure that you check the Validation Required? box.

a. Synchronize the extension behavior for the button. For more information about synchronizing the extension behavior, see Chapter 6, "Synchronizing New and Existing UI Resources".

b. Start the Eclipse SDK.

c. In the navigator view, expand the plug-in project that contains this screen.

d. Expand the package and open the extension behavior class that you specified when synchronizing the extension behavior. For more information about synchronizing the extension behavior, see Chapter 6, "Synchronizing New and Existing UI Resources".

e. In the validateButtonClick() method, add logic to provide the desired implementation for the button click event.

3.3 Adding a Checkbox

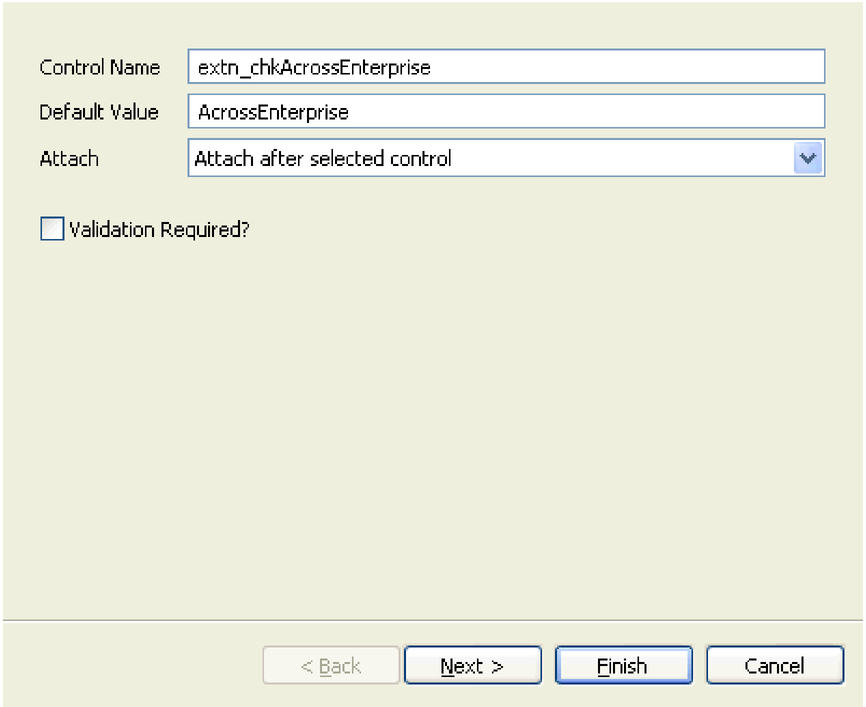
To add a check box:

1. Start the Rich Client Platform Extensibility Tool.
2. Click 

. Do any of the following to add a checkbox:

   - Select the control where you want to add the checkbox and click once.
   - Select the composite or group where you want to add the checkbox and click once.

   The Add Checkbox window displays. For field value descriptions, see Table 3–5.



Note: You can place the checkbox either in the row layout container or grid layout container. However, the row layout container does not have the layout data. Therefore, if you place the checkbox in the row layout container, the Layout Setup Page is disabled.

---

3. Click Finish to add the new checkbox with the default layout data, theme, and null binding attributes.
4. Click Next to specify the layout data options. For field value descriptions, see Table 3–2.
5. Click Finish to add the new checkbox with the default theme and null binding attributes.
6. Click Next to specify the binding options.

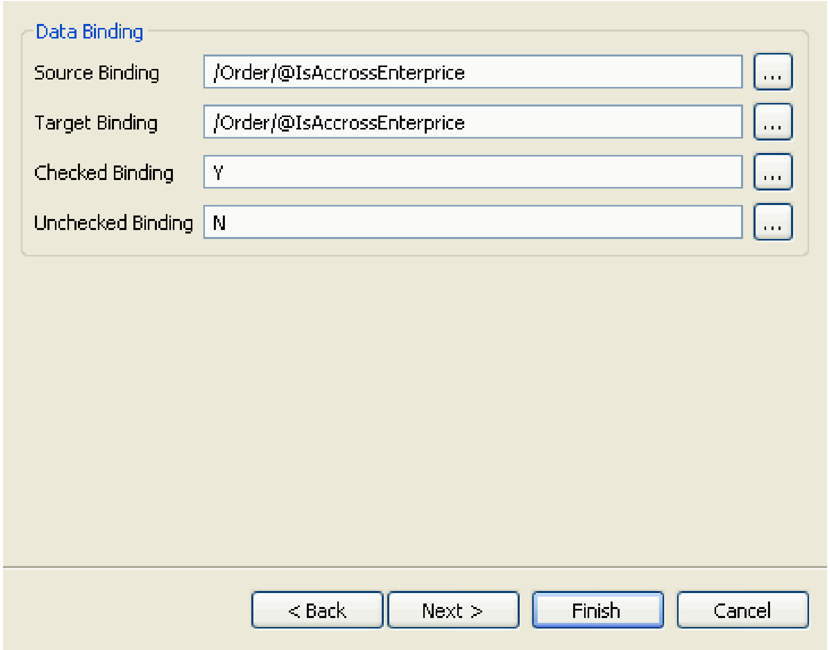

Data Binding
Source Binding
/Order/@IsAccrossEnterprice
Target Binding
/Order/@IsAccrossEnterprice
Checked Binding
Y
Unchecked Binding
N
< Back
Next >
Finish
Cancel

Table 3–7 Add Checkbox, Binding Options Page

Field	Description
Source Binding	Enter the XML path to populate the new field, if applicable. Click ... to view the available XML paths. The Source Bindings Tree pop-up window displays. Select the appropriate XML path from the tree. You can also press Ctrl+Space and select the appropriate XML path from the drop-down list.
Target Binding	Enter the XML path to send data to the API from the new field, if applicable. You can specify multiple target bindings by using a semicolon. Click ... to view the available XML paths. The Target Bindings Tree pop-up window displays. Select the appropriate XML path from the tree. You can also press Ctrl+Space and select the appropriate XML path from the drop-down list.
Checked Binding	Enter the XML path to specify the checked binding attribute, if applicable. Click ... to view the available XML paths. The Checked Bindings Tree pop-up window displays. Select the appropriate XML path from the tree. You can also press Ctrl+Space and select the appropriate XML path from the drop-down list.
Unchecked Binding	Enter the XML path to specify the unchecked binding attribute, if applicable. Click ... to view the available XML paths. The Unchecked Bindings Tree pop-up window displays. Select the appropriate XML path from the tree. You can also press Ctrl+Space and select the appropriate XML path from the drop-down list.

7. Click Finish to add the new checkbox with the default theme binding.
8. Click Next to specify the theme binding. For field value descriptions, see Table 3–4.
9. Click Finish.

10. Click to save the changes made to the extension file.

11. If you have specified the default value, bindings, or theme entry for the checkbox, you must synchronize the resource files. For more information about synchronizing resource files, see Chapter 6, "Synchronizing New and Existing UI Resources".

3.4 Adding a Radio Button

To add a radio button:

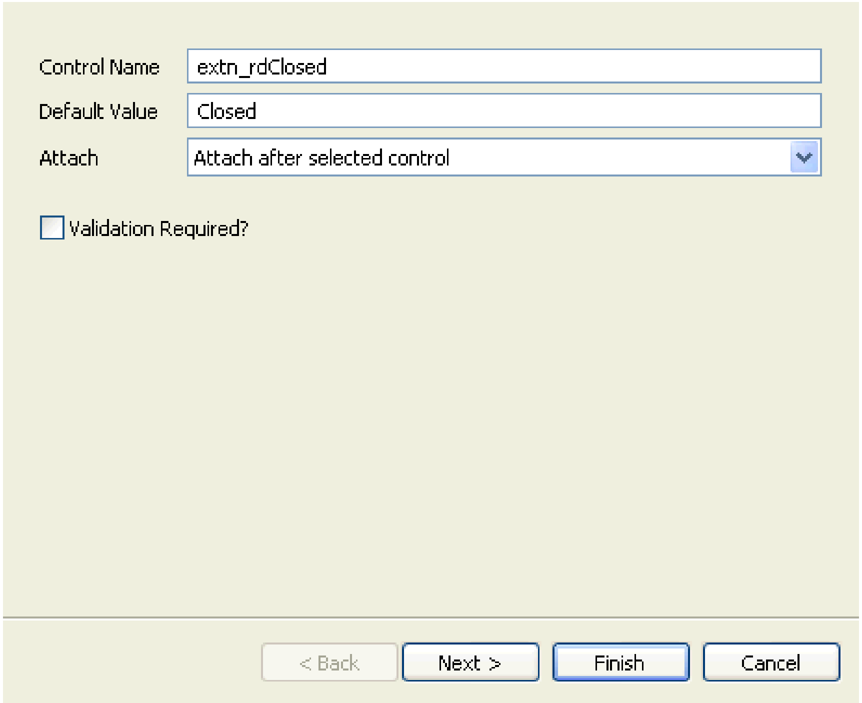
1. Start the Rich Client Platform Extensibility Tool.

2. Click . Do any of the following to add a radio button:

   - Select the control where you want to add the radio button and click once.

   - Select the composite or group where you want to add the radio button and click once.

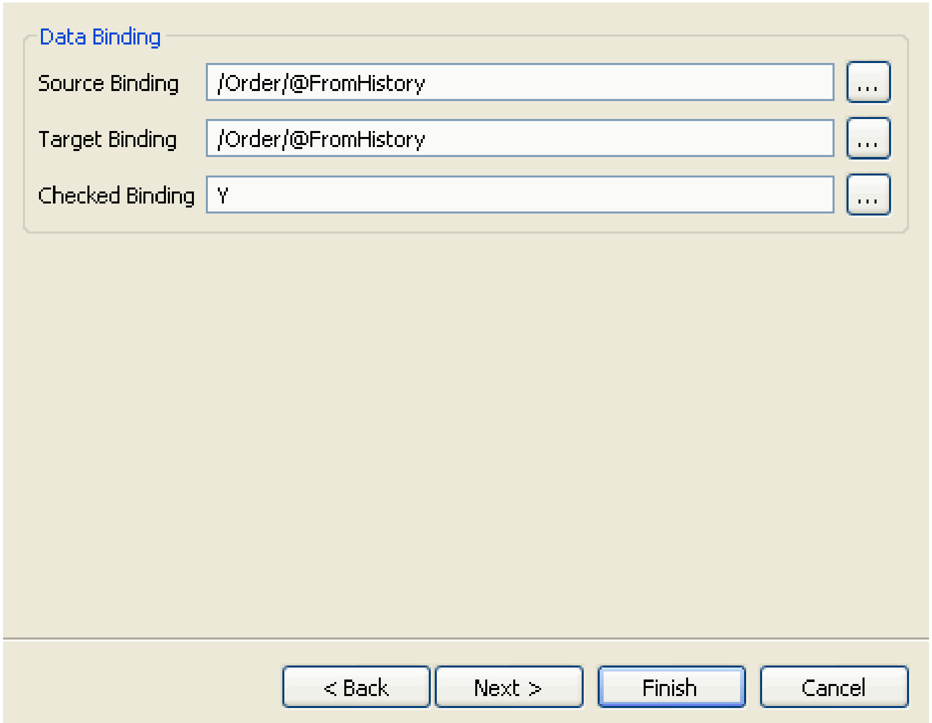
   The Add Radio Button window displays. For field value descriptions, see Table 3–5.



Note: You can place the radio button either in the row layout container or grid layout container. However, the row layout container does not have the layout data. Therefore, if you place the radio button in the row layout container, the Layout Setup Page is disabled.

---

3. Click Finish to add the new radio button with the default layout data, theme, and null binding attributes.
4. Click Next to specify the layout data options. For field value descriptions, see Table 3–2.
5. Click Finish to add the new radio button with the default theme and null binding attributes.
6. Click Next to specify the binding options. For field value descriptions, see Table 3–7.



7. Click Finish to add the new radio button with the default theme binding.
8. Click Next to specify the theme binding. For field value descriptions, see Table 3–4.
9. Click Finish.
10. Click  to save the changes made to the extension file.
11. If you have specified the default value, bindings, or theme entry for the radio button, you must synchronize the resource files. For more information about synchronizing resource files, see Chapter 6, "Synchronizing New and Existing UI Resources".

3.5 Adding a Text Box

To add a text box:

1. Start the Rich Client Platform Extensibility Tool.
2. Click Text Box . Do any of the following to add a text box:
   - Select the control where you want to add the text box and click once.
   - Select the composite or group where you want to add the text box and click once.

   The Add Text Box window displays.

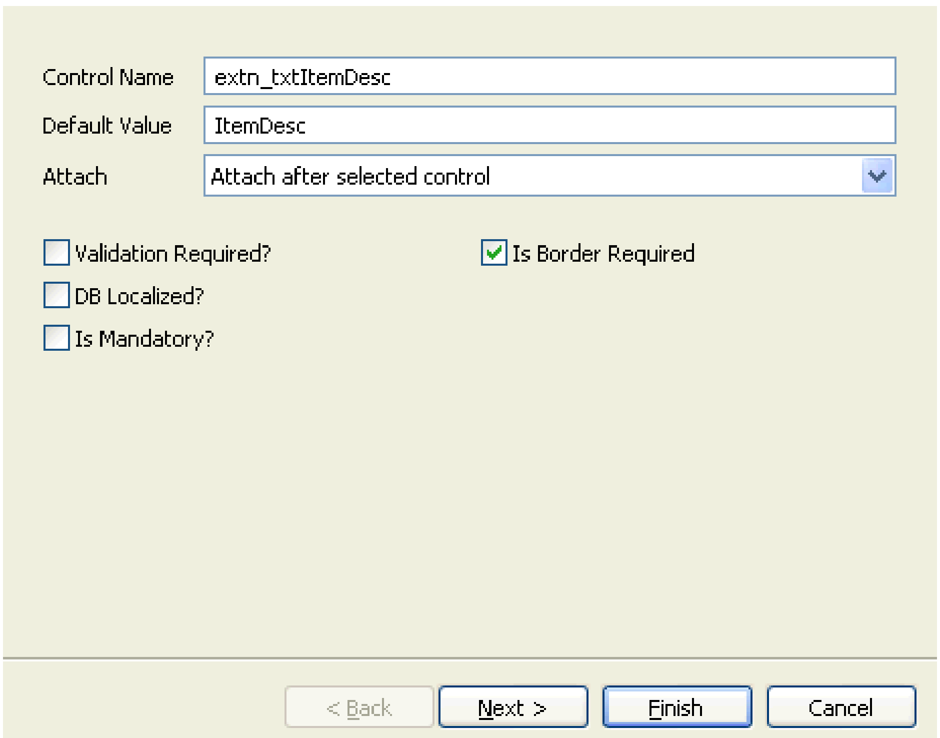
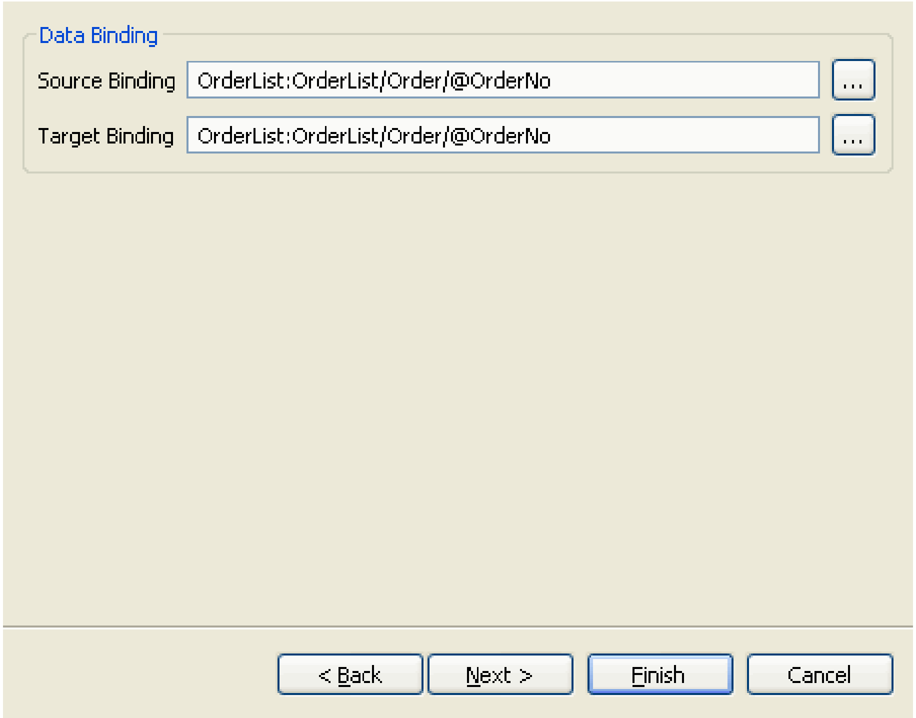


Table 3–8 Add New Text Box

Field	Description
Control Name	Mandatory field. Enter a unique control name for the new control. Every field on the form must have a unique logical name for reference. (Optional) Prefix the control name with "extn_". If you do not specify this, the system automatically adds "extn_" to the control name.
Default Value	Mandatory field. Enter the default value to display on the screen.
Attach	Select the appropriate value from the drop-down list. Valid values are "Attach after selected control", "Attach before selected control", and "Attach within selected control". **Note:** If you are adding the new field to an empty composite or group, by default, the "Attach within selected control" value displays.
Validation Required?	Check this box to validate a new field, if applicable. The Rich Client Platform invokes the validation method on the extended behavior class of this form when the field loses focus.
DB Localized?	Check this box to retrieve data for the new field from a localized database (DB) column, if applicable.
Is Mandatory?	Check this box if you want to make this field mandatory.
Is Border Required?	Uncheck this box if you want to remove the border around the field.

3. Click Finish to add the text box with the default layout data, theme, and null binding attributes.
4. Click Next to specify the layout data options. For field value descriptions, see Table 3–2.
5. Click Finish to add the text box with the default theme and null binding attributes.

6. Click Next to specify the binding options. For field value descriptions, see Table 3–7.



7. Click Finish to add the text box with the default theme binding.
8. Click Next to specify the theme binding. For field value descriptions, see Table 3–4.
9. Click Finish.
10. Click  to save the changes made to the extension file.
11. If you have specified the default value, bindings, or theme entry for the text box, you must synchronize the resource files. For more information about synchronizing resource files, see Chapter 6, "Synchronizing New and Existing UI Resources".

3.6 Adding StyledText Component

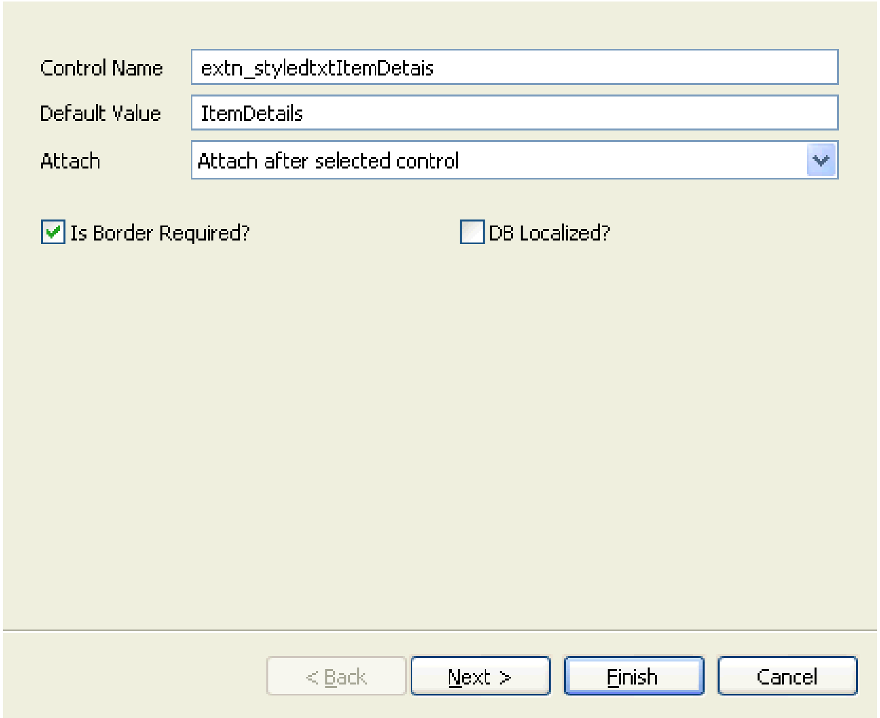
The StyledText component is a customized control that can be used to display and edit text with different colors and font styles. The StyledText components are powerful and multifaceted components suitable for high-end needs and offer more avenues for customization in comparison to other text components.

To add a styled text:

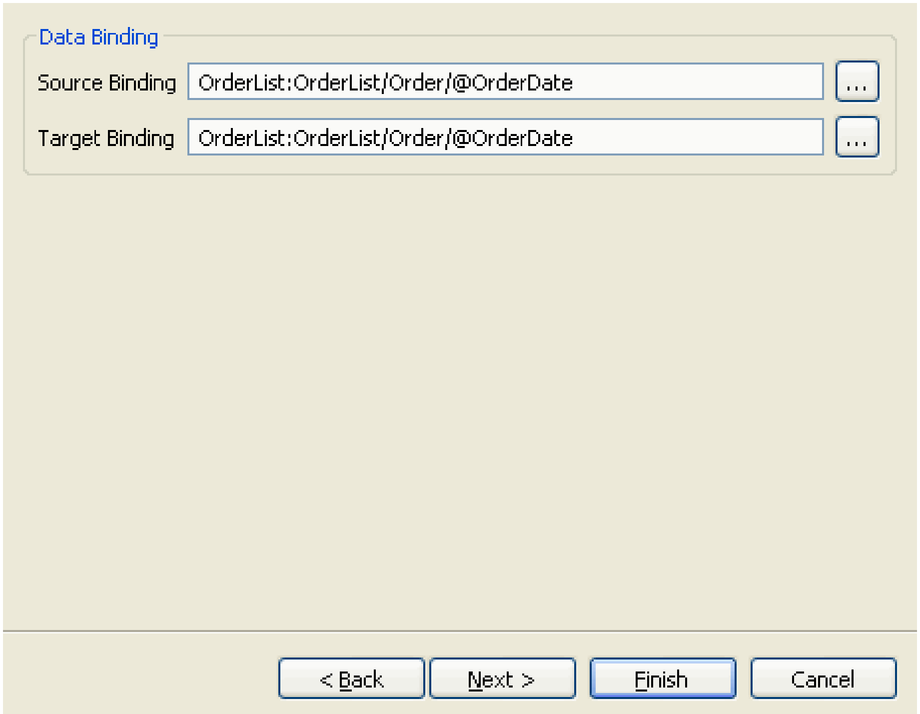
1. Start the Rich Client Platform Extensibility Tool.
2. Click 

. Do any of the following to add a styled text:
    - Select the control where you want to add the styled text and click once.
    - Select the composite or group where you want to add the styled text and click once.

    The Add Styled Text window displays. For field value descriptions, see Table 3–8.



3. Click Finish to add the new styled text with the default layout data, theme, and null binding attributes.
4. Click Next to specify the layout data options. For field value descriptions, see Table 3–2.
5. Click Finish to add the styled text with the default theme and null binding attributes.
6. Click Next to specify the binding options. For field value descriptions, see Table 3–7.



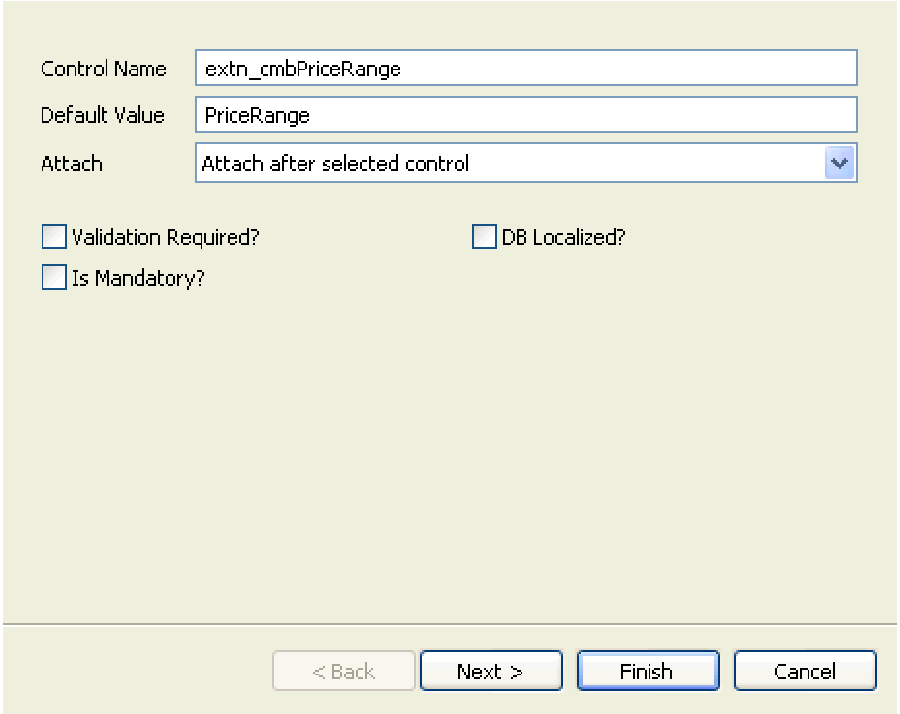
7. Click Finish to add the styled text with the default theme binding.
8. Click Next to specify the theme binding. For field value descriptions, see Table 3–4.
9. Click Finish.
10. Click  to save the changes made to the extension file.
11. If you have specified the default value, bindings, or theme entry for the styled text, you must synchronize the resource files. For more information about synchronizing resource files, see Chapter 6, "Synchronizing New and Existing UI Resources".

3.7 Adding a Combo Box

To add a combo box:

1. Start the Rich Client Platform Extensibility Tool.
2. Click Combo Box. Do any of the following to add a combo box:
   - Select the control where you want to add the combo box and click once.
   - Select the composite or group where you want to add the combo box and click once.

   The Add Combo Box window displays. For field value descriptions, see Table 3–8.



3. Click Finish to add the new combo box with the default layout data, theme, and null binding attributes.
4. Click Next to specify the layout data options. For field value descriptions, see Table 3–2.
5. Click Finish to add the new combo box with the default theme and null binding attributes.

6. Click Next to specify the binding options.

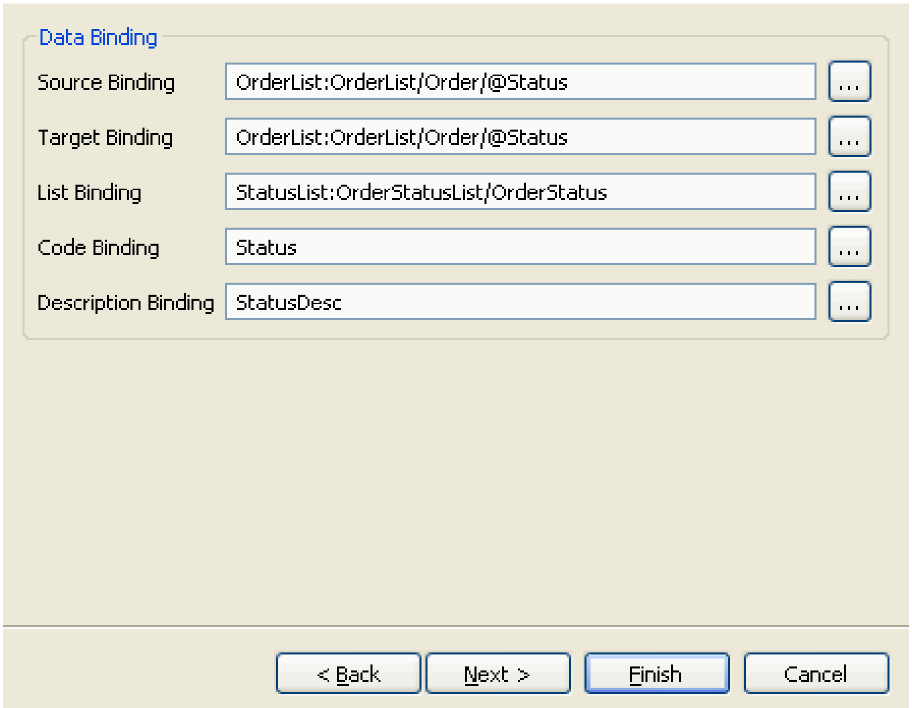


Table 3–9 Add Combo Box, Binding Options Page, Data Binding

Field	Description
Source Binding	Enter the XML path to populate the new field, if applicable. Click  to view the available XML paths. The Source Bindings Tree pop-up window displays. Select the appropriate XML path from the tree. You can also press Ctrl+Space and select the appropriate XML path from the drop-down list.
Target Binding	Enter the XML path to send data to the API for the new field, if applicable. You can specify multiple target bindings by using a semicolon. Click  to view the available XML paths. The Target Bindings Tree pop-up window displays. Select the appropriate XML path from the tree. You can also press Ctrl+Space and select the appropriate XML path from the drop-down list.
List Binding	Enter the XML path to get a list of items for the new field. Click  to view the available XML paths. The List Bindings Tree pop-up window displays. Select the appropriate XML path from the tree. You can also press Ctrl+Space and select the appropriate XML path from the drop-down list.

Table 3–9 Add Combo Box, Binding Options Page, Data Binding

Field	Description
Code Binding	Enter the XML path to send the value of an attribute to the API for the new field. Click ... to view the available XML paths. The Code Bindings Tree pop-up window displays. Select the appropriate XML path from the tree. You can also press Ctrl+Space and select the appropriate XML path from the drop-down list.
Description Binding	Enter the XML path to specify the value to display on the screen. Click ... to view the available XML paths. The Description Bindings Tree pop-up window displays. Select the appropriate XML path from the tree. You can also press Ctrl+Space and select the appropriate XML path from the drop-down list.

7. Click Finish to add the new combo box with the default theme binding.
8. Click Next to specify the theme binding. For field value descriptions, see Table 3–4.
9. Click Finish.
10. Click on save the changes made to the extension file.
11. If you have specified the default value, bindings, or theme entry for the combo box, you must synchronize the resource files. For more information about synchronizing resource files, see Chapter 6, "Synchronizing New and Existing UI Resources".

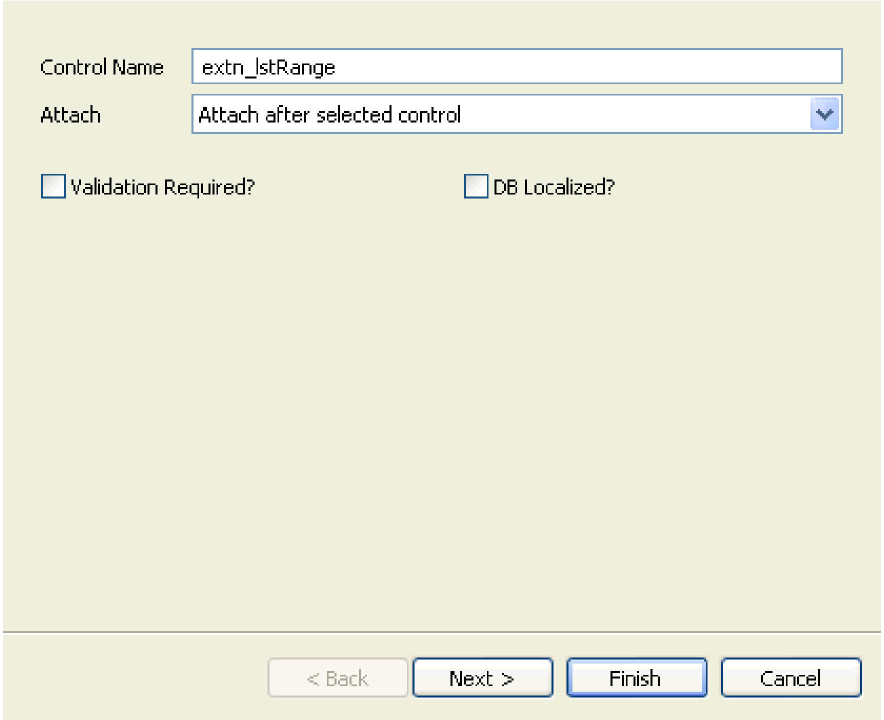
3.8 Adding a List Box

To add a list box:

1. Start the Rich Client Platform Extensibility Tool.
2. Click List Box. Do any of the following to add a list box:
   - Select the control where you want to add the list box and click once.

- Select the composite or group where you want to add the list box and click once.

The Add List Box window displays. For field value descriptions, see Table 3–8.



3. Click Finish to add the list box with the default layout data, theme, and null binding attributes.
4. Click Next to specify the layout data options. For field value descriptions, see Table 3–2.
5. Click Finish to add the list box with the default theme and null binding attributes.
6. Click Next to specify the binding options. For field value descriptions, see Table 3–9.
7. Click Finish to add the list box with the default theme binding.

8. Click Next to specify the theme binding. For field value descriptions, see Table 3–4.
9. Click Finish.
10. Click  to save the changes made to the extension file.
11. If you have specified the default value, bindings, or theme entry for the list box, you must synchronize the resource files. For more information about synchronizing resource files, see Chapter 6, "Synchronizing New and Existing UI Resources".
12. Make sure that you create the extension behavior for the screen on which you are adding this custom control. Also, you need to override the getBindingdata() method to return the custom control binding data object. This method is called by Rich Client Platform while creating the custom control.

3.9 Adding a Standard Or Advanced Table Column

You can add either a standard column or advanced column to a table.

To add a standard or advanced column to a table:

1. Start the Rich Client Platform Extensibility Tool.
2. Click Table Column. Select the column where you want to add the new standard or advanced column and click once.

   The Add Table Column window displays.

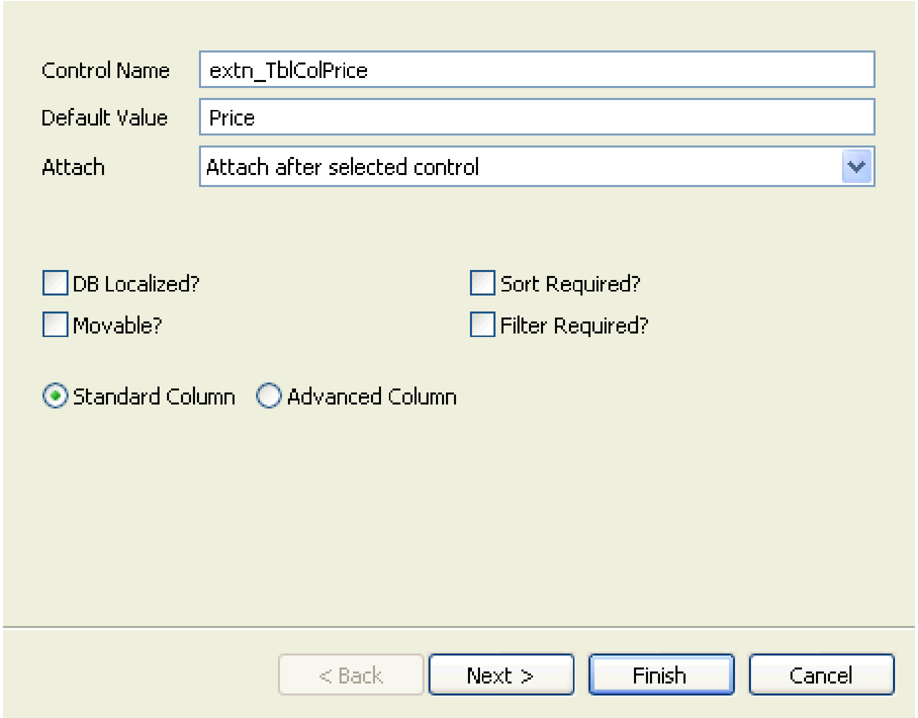


Table 3–10 Add Table Column

Field	Description
Control Name	Mandatory field. Enter a unique control name for the new column. Every field on the form must have a unique logical name for reference. (Optional) Prefix the control name with "extn_". If you do not specify this, the system automatically adds "extn_" to the control name.
Default Value	Mandatory field. Enter the default value to display on the screen.
Attach	Select the appropriate value from the drop-down list. Valid values are "Attach after selected control" and "Attach before selected control".

Table 3–10 Add Table Column

Field	Description
DB Localized?	Check this box if you want to retrieve data for this column from a localized DB column.
Movable?	Check this box to move the column, if applicable.
Sort Required?	Check this box to sort data in a particular order, if applicable.
Filter Required?	Check this box to filter data, if applicable.
Standard Column	Choose this if you want to add a standard column.
Advanced Column	Choose this if you want to add an advanced column.

3. Click Finish to add the new column with the default column layout data, theme, and null binding attributes.
4. Click Next to specify the column layout options.

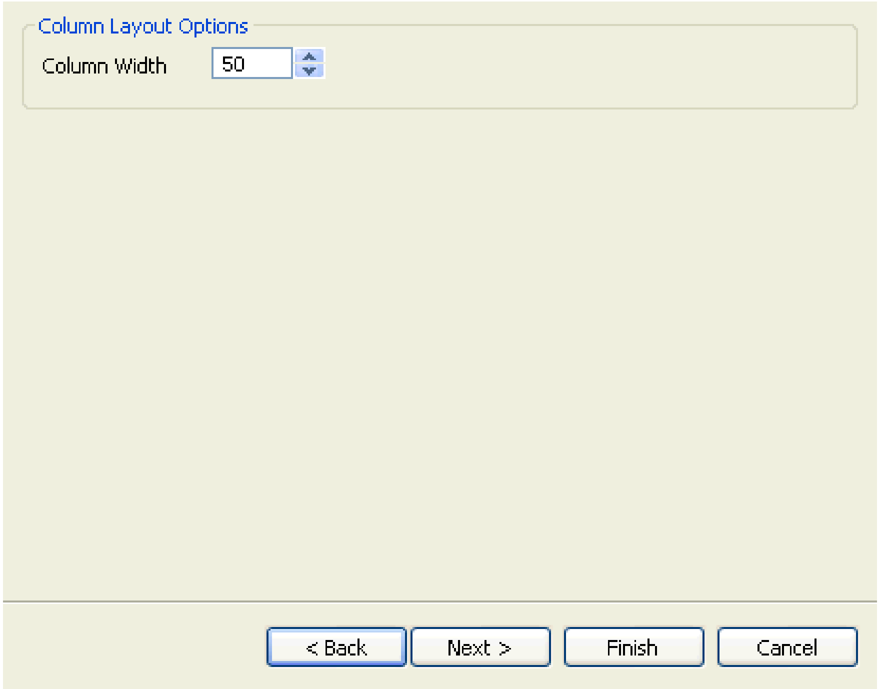


Table 3–11 Add Table Column, Layout Setup Page

Field	Description
Column Width	Increase or decrease the width of the column, if necessary.

5. Click Finish to add the column with the default theme and null binding attributes.
6. Click Next to specify the binding options.

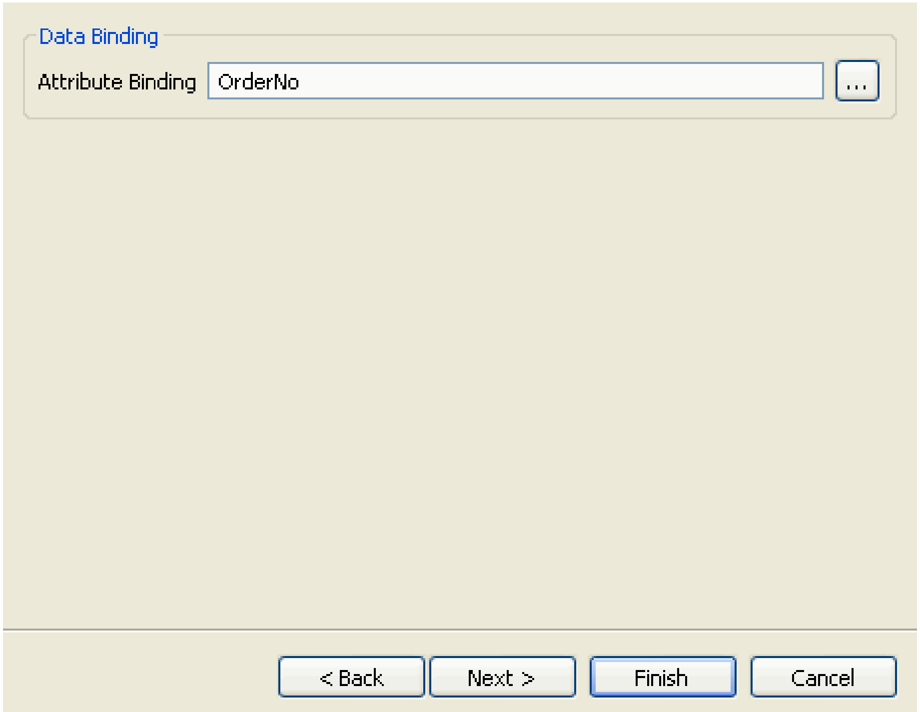


Table 3–12 Add Table Column, Binding Options Page

Field	Description
Attribute Binding	Enter the XML path to specify the binding attribute, if applicable. Click  to view the available XML paths. The Attribute Bindings Tree pop-up window displays. Select the appropriate XML path from the tree. You can also press Ctrl+Space and select the appropriate XML path from the drop-down list. **Note:** If you are adding an advanced column, this field is disabled. You cannot specify the attribute binding for an advanced column using the Rich Client Platform Extensibility Tool.

7. Click Finish to add the new column with the default theme binding.

8. Click Next to specify the theme binding.

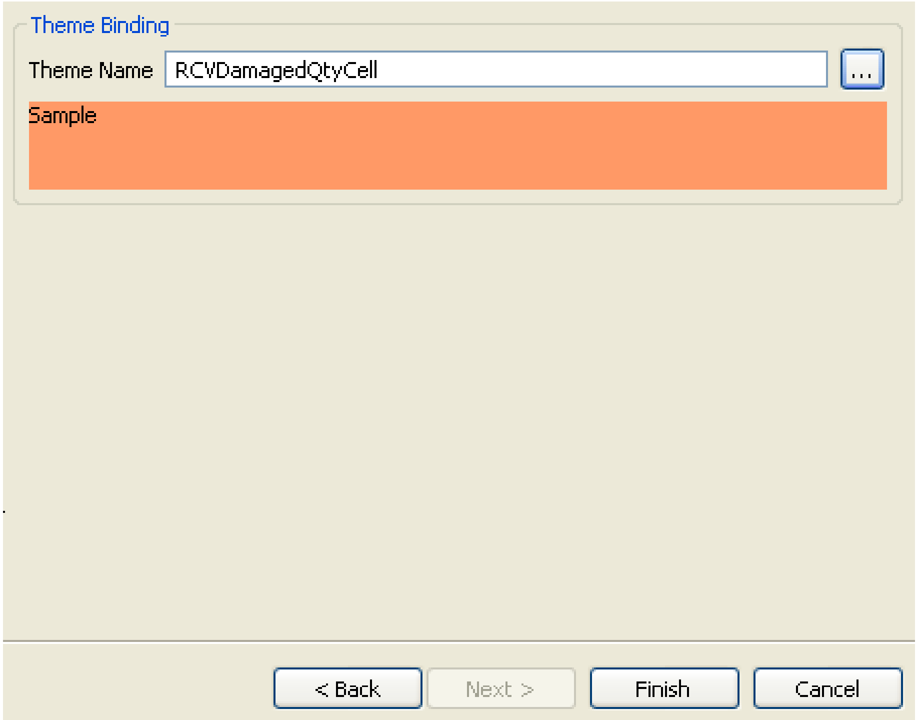


Table 3–13 Add Table Column

Field	Description
Theme Name	Enter the theme name you want to apply for the new field, if applicable. Click [...] to view the available themes. The Themes Tree pop-up window is displayed, which lists all the available themes and also provides a preview of each theme. Select the appropriate theme name you want to apply. You can also press Ctrl+Space and select the appropriate theme name you want to apply from the drop-down list. The current theme applied to the selected control is displayed below the theme name. If an image is specified as the theme, then the image is displayed. **Note:** If you are adding an advanced column, this field is disabled. You cannot specify the theme binding for an advanced column using the Rich Client Platform Extensibility Tool.

9. Click Finish.
10. Click to save the changes made to the extension file.
11. If you have specified the default value, bindings, or theme entry for the table column, you must synchronize the resource files. For more information about synchronizing resource files, see Chapter 6, "Synchronizing New and Existing UI Resources".

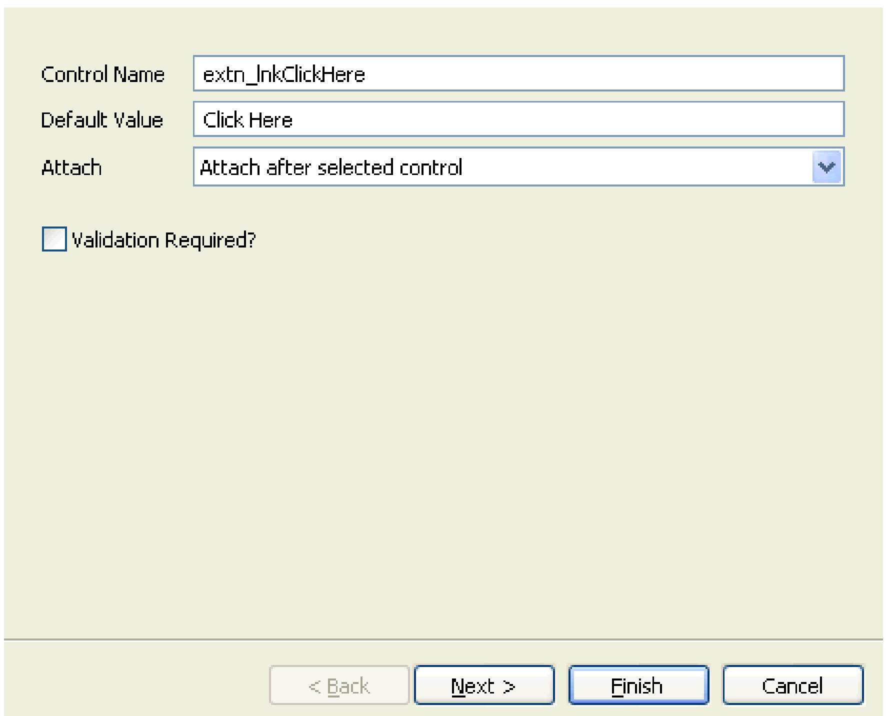
> **Note:** If you add an advanced column, create and synchronize the appropriate extension behavior for setting the bindings for the extended table. For more information about creating extension behavior, see Section 4.2, "Creating Extension Behavior". For more information about synchronizing extension behavior, see Chapter 6, "Synchronizing New and Existing UI Resources".
>
> After creating and synchronizing the extension behavior, set the bindings for the extended table and advanced column you added in the extension behavior class.

3.10 Adding a Link

To add a link:

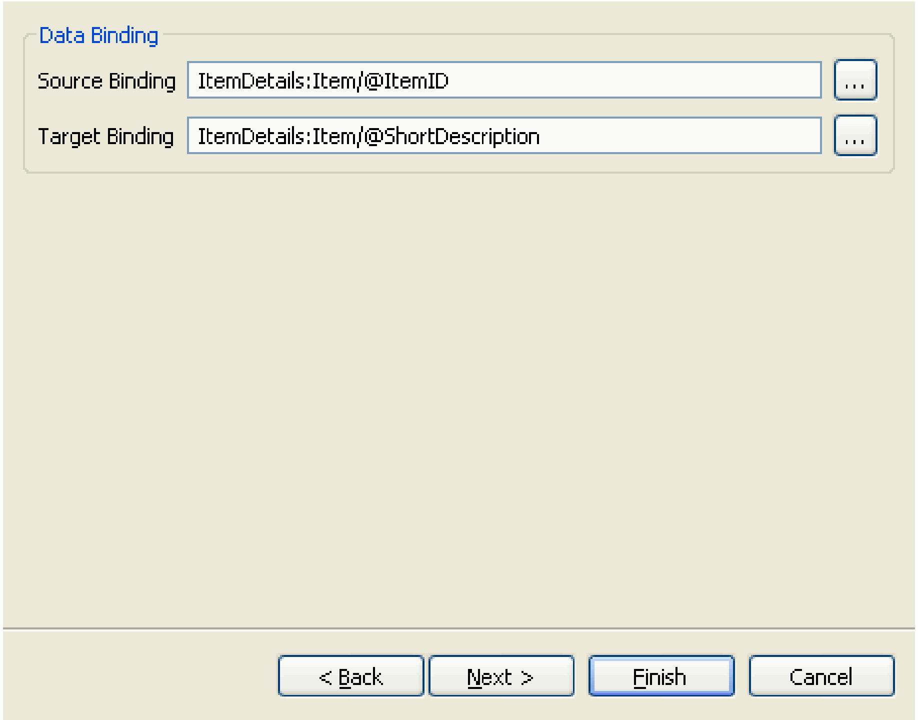
1. Start the Rich Client Platform Extensibility Tool.
2. Click Link . Do any of the following to add a link:
    - Select the control where you want to add the link and click once.
    - Select the composite or group where you want to add the link and click once.

   The Add Link window displays. For field value descriptions, see Table 3–5.



> **Note:** You can place the link either in the row layout container or grid layout container. However, the row layout container does not have the layout data. Therefore, if you place the link in the row layout container, the Layout Setup Page is disabled.

3. Click Finish to add the new link with the default layout data, theme, null binding attributes, and empty hot key binding.
4. Click Next to specify the layout data options. For field value descriptions, see Table 3–2.
5. Click Finish to add the link with the default theme, null binding attributes, and empty hot key binding.
6. Click Next to specify the binding options. For field value descriptions, see Table 3–5.



7. Click Finish to add the link with the default theme binding and empty hot key binding.
8. Click Next to specify the theme binding. For field value descriptions, see Table 3–4.
9. Click Finish to add the link with the empty hot key binding.
10. Click Next to specify hot key binding. For field value descriptions, see Table 3–6.
11. Click Finish.

    If you have specified the hot key binding, the Restart Application pop-up window displays and prompts you to save the extension file and restart the application to view the changes.

> **Note:** Whenever you specify the hot key binding for a control, you must save the extension file and restart the application to view the changes of the hot key binding. Make sure that you clear the configuration data before restarting the application.

12. Click  to save the changes made to the extension file.
13. If you have specified the default value, bindings, or theme entry for the link, you must synchronize the resource files. For more information about synchronizing resource files, see Chapter 6, "Synchronizing New and Existing UI Resources".

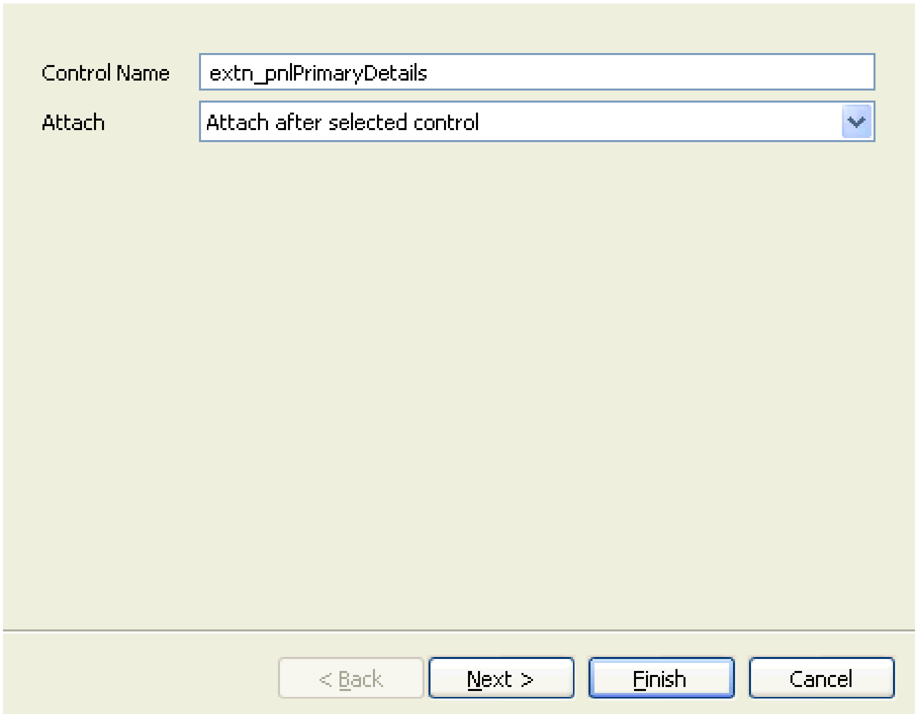
3.11 Adding a Composite

To add a composite:

1. Start the Rich Client Platform Extensibility Tool.
2. Click 

. Do any of the following to add a composite:
   - Select the control where you want to add the composite and click once.
   - Select the composite or group where you want to add the composite and click once.

   The Add Composite window displays. For field value descriptions, see Table 3–1.



3. Click Finish to add the composite with the default layout data.
4. Click Next to specify the layout data options. For field value descriptions, see Table 3–2.
5. Enter information in the applicable fields and click Finish.
6. Click to save the changes made to the extension file.

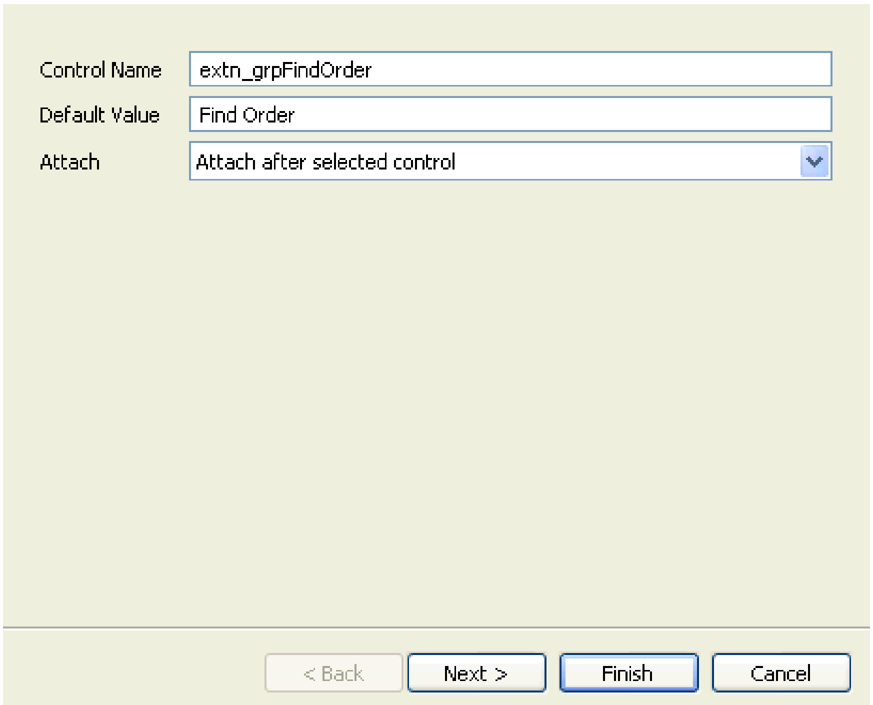
3.12 Adding a Group

To add a group:

1. Start the Rich Client Platform Extensibility Tool.
2. Click Group . Do any of the following to add a group:
   - Select the control where you want to add the group and click once.

- Select the composite or group where you want to add the group and click once.

The Add Group window displays. For field value descriptions, see Table 3–1.



3. Click Finish to add the group with the default layout data.
4. Click Next to specify the layout data options. For field value descriptions, see Table 3–2.
5. Click Finish.
6. Click to save the changes made to the extension file.
7. If you have specified the default value for the new group, you must synchronize the resource files. For more information about synchronizing resource files, see Chapter 6, "Synchronizing New and Existing UI Resources".

3.13 Adding an External Panel

To add an external panel:

1. Start the Rich Client Platform Extensibility Tool.
2. Click External Panel. Do any of the following to add a group:
   - Select the control where you want to add the external panel and click once.
   - Select the composite or group where you want to add the external panel and click once.

   The Add External Panel window displays.

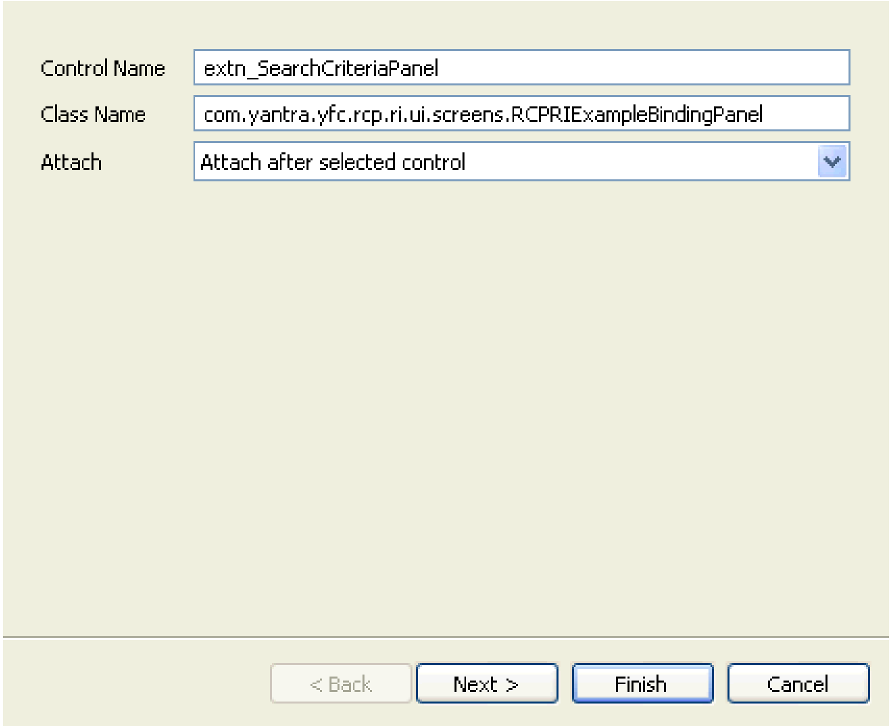


Table 3–14 Add External Panel

Field	Description
Control Name	Mandatory field. Enter a unique control name for the external panel. Every field on the form must have a unique logical name for reference. (Optional) Prefix the control name with "extn_". If you do not specify this, the system automatically adds "extn_" to the control name.
Class Name	Mandatory field. Enter the external panel's class name including the package name. For example, com.yantra.yfc.rcp.ri.ui.screens.RCPRIExampleBindingPanel. Here com.yantra.yfc.rcp.ri.ui.screens is the package name and RCPRIExampleBindingPanel is the external panel's class name.
Attach	Select the appropriate value from the drop-down list. Valid values are "Attach after selected control" and "Attach before selected control".

3. Click Finish to add the group with the default layout data.
4. Click Next to specify the layout data options. For field value descriptions, see Table 3–2.
5. Click Finish.
6. Click  to save the changes made to the extension file.
7. Open the external panel's class in the Java Editor.
8. In the class, create the constructor as following:

```
public RCPDRIBindingPanel1(Composite parent, int style, YRCExtentionBehavior
behavior) {
}
```

Note: Make sure that the constructor contains only the following arguments and in this specific order:

- Composite parent, int style, and YRCExtensionBehavior behavior.

3.14 Moving Controls and Table Columns

Using the move option you can move or rearrange controls on the screen.

You can move:

- Individual controld on the screen.
- A complete composite or group containing one or more controls

  All child controls of the composite or group moves along with the group.
- Individual columns in the table.

Note: If a screen comprises two independent or reusable forms, the Rich Client Platform Extensibility Tool does not allow you to move a control from one form to another.

3.14.1 Moving a Control

To move a control:

1. Start the Rich Client Platform Extensibility Tool.
2. Click 

. Select the control you want to move.

Note: If you attempt to move controls for which a unique name is not set, an error message displays.

3. Click the target control where you want to move the selected control. The Move Field window displays.

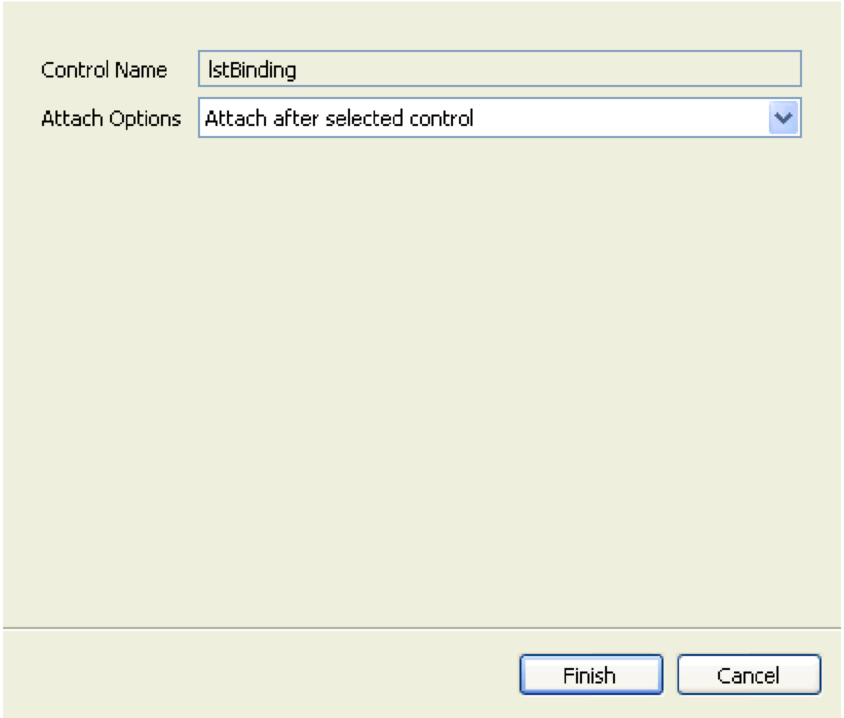


Table 3–15 Move Existing Field

Field	Description
Control Name	Disabled field. Displays the name of the selected field you want to move. Every field on the form must have a unique logical name used only for reference.
Attach Options	Select the appropriate value from the drop-down list. Valid values are "Attach after selected control" and "Attach before selected control". **Note:** If you are moving the selected field to an empty composite or group, the "Attach within selected control" value displays.

4. Click Finish.

3.14.2 Moving Table Columns

You can rearrange columns in a table. However, you cannot move a column from one table to another table.

> **Note:** You cannot move a table column after or before a newly added column.

To move a table column:

1. Start the Rich Client Platform Extensibility Tool.
2. Click 

. Select the table column you want to move.

> **Note:** Set a unique name for all columns in a table. Otherwise, an error message displays.

3. Click the target table column where you want to move the selected table column. The Move Table Column window displays.

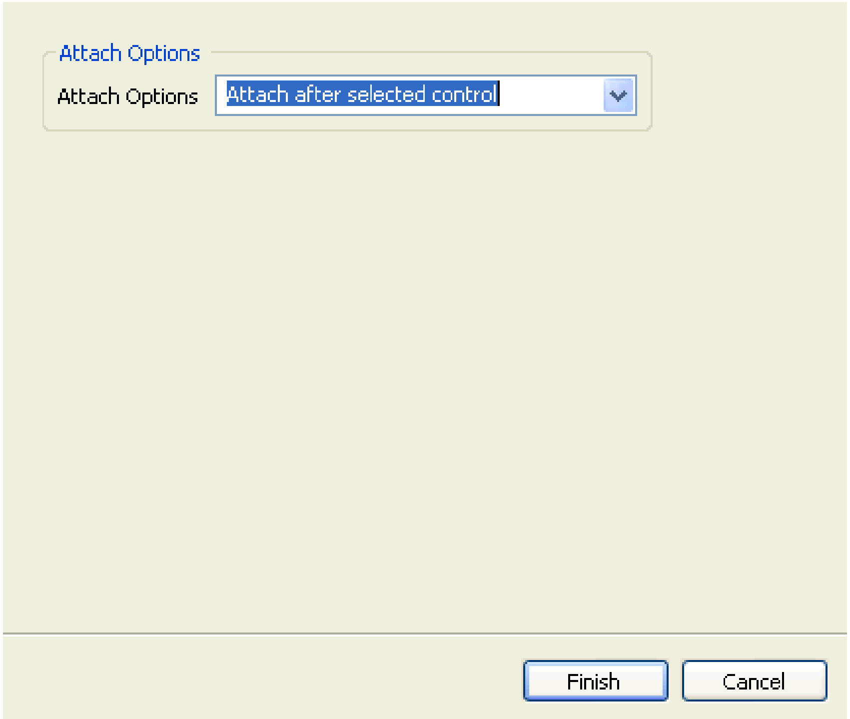


Table 3–16 Move Table Column

Field	Description
Attach Options	Select the appropriate value from the drop-down list. Valid values are "Attach after selected control" and "Attach before selected control".

4. Click Finish and reopen the screen to view the changes made to the table.

4

Related Tasks, Extension Behavior, and Hot Keys in the Rich Client Platform UI

4.1 Adding Related Tasks

Using the Related Task option of the Rich Client Platform Extensibility Tool, you can add new related tasks for a task that is open in an editor. You must define the group and category for each related task. All the related tasks can belong to multiple categories, but are limited to one group.

Prior to adding a new related task, you must be aware of the following information:

- Category Information—When a new related task is associated with the active task running in the current editor, you must know the categories in which the active task is included so that the new related task can be displayed under the Related Tasks view.
- Group Information—To display a new related task, you can either use an existing group or create a new group.

To add a new related task:

1. Start the Rich Client Platform Extensibility Tool.
2. Navigate to the task you want to extend in the Rich Client Platform application.
3. Click 

and click once on the form. The Add Related Task window displays.

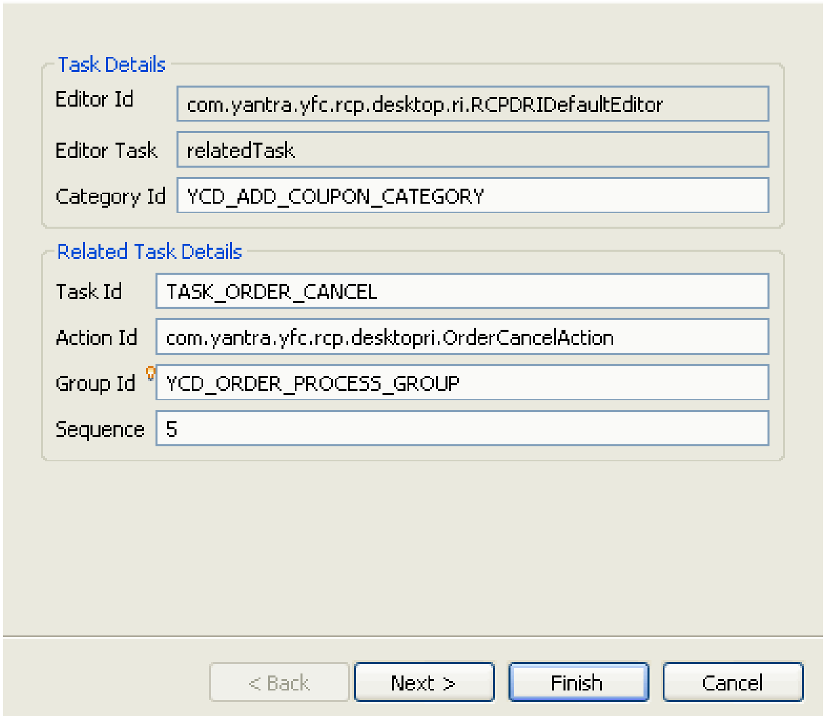


Table 4–1 Add Related Task

Field	Description
Task Details	
Editor Id	Disabled field. Displays the identifier of the currently opened editor on the screen.
Editor Task	Disabled field. Displays the identifier of the task that is currently open in the editor.
Category Id	Enter the identifier of the category in which the new related task is included. You can add a related task to an existing category or new category. You can also press Ctrl+Space to select the category identifier from the list of available categories.

Table 4–1 Add Related Task

Field	Description
Related Task Details	
Task Id	Enter a unique identifier for the new related task you want to add.
Action Id	Enter the identifier of the action you want to invoke upon clicking the related task. You can also press Ctrl+Space to select the action identifier from the list of available actions.
Group Id	Enter the identifier of the group in which you want to display the related task so that the new related task can be displayed in the Related Tasks view. You can add a related task in an existing group or a new group. You can also press Ctrl+Space to select the group identifier from the list of available groups.
Sequence	Enter the sequence number to display the new group in a particular order, if applicable. The groups are displayed in ascending order by their sequence number. **Note:** This field is enabled only if you specify a new group identifier to display the related task.

4. Click Next to set permissions for the new related task.

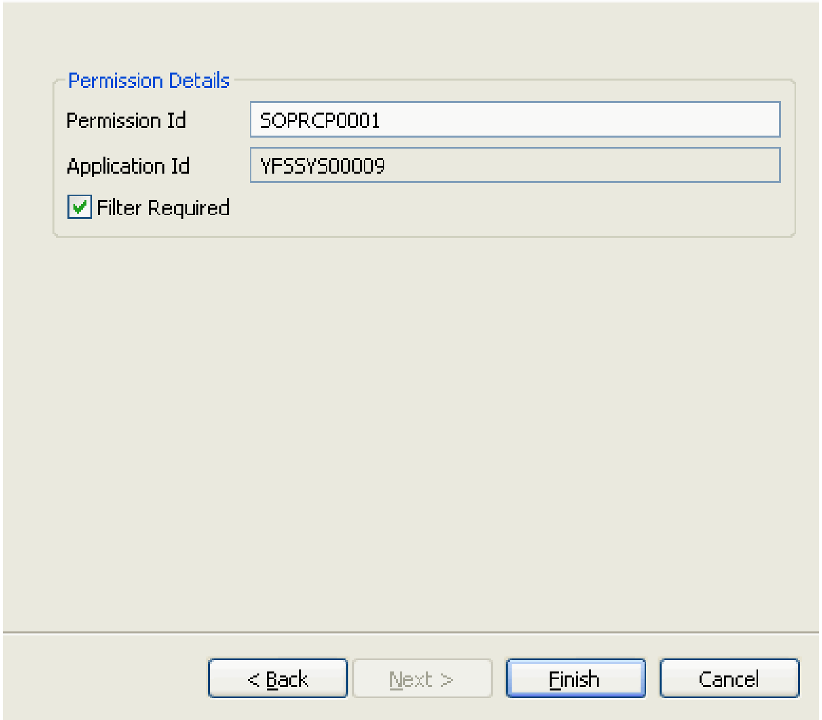


Table 4–2 Add Related Task, Related Task Permissions

Field	Description
Permission Id	Enter the resource identifier.
Application Id	Enter the identifier of the application for which you are adding the related task.
Filter Required	Check this box if you want to filter the related task depending on the custom criteria.

5. Click Finish to view the newly added related task in the Related Task view.

Note: You must synchronize files after adding the new related task in order to add them to the `plugin.xml` file. The name of the related task and the group are based on bundle entries defined for the corresponding identifiers.

---

6. Synchronize the resource files for the changes you made to the new related task. For more information about synchronizing resource files, see Chapter 6, "Synchronizing New and Existing UI Resources".

4.2 Creating Extension Behavior

You must create the extension behavior whenever you are extending an existing wizard or screen:

- To perform custom validations for the newly added fields
- To call an API or Service
- To set the extension model to populate a screen or wizard page
- To set bindings for the advanced columns you add through the Rich Client Platform Extensibility Tool.

If you are extending a screen which is not a wizard, the tool creates the normal extension behavior and if you are extending a screen which is a wizard, the tool creates the wizard extension behavior.

To create extension behavior:

1. Start the Rich Client Platform Extensibility Tool.
2. Click Extension Behavior and click on the form. The Create Extension Behavior? message displays. The message box prompts you to confirm whether you want to create an extension behavior for the current form. The message box also displays the identifier of the current form.
3. Click OK.
4. Synchronize the extension behavior that you created to specify the implementation class, if applicable. For more information about synchronizing extension behavior, see Chapter 6, "Synchronizing New and Existing UI Resources".

4.3 Configuring Hot Keys

Using the Rich Client Platform Extensibility Tool, you can configure the hot keys to perform a specific task or operation.

To configure the hot keys:

1. Start the Rich Client Platform Extensibility Tool.
2. Click Configure Hot Keys and click once on the screen. The Hot Key Configuration window displays.

   By default, in the ContextId field, the context identifier of the current screen displays. A list of hot keys defined in the global context for the defaulted context identifier also displays.

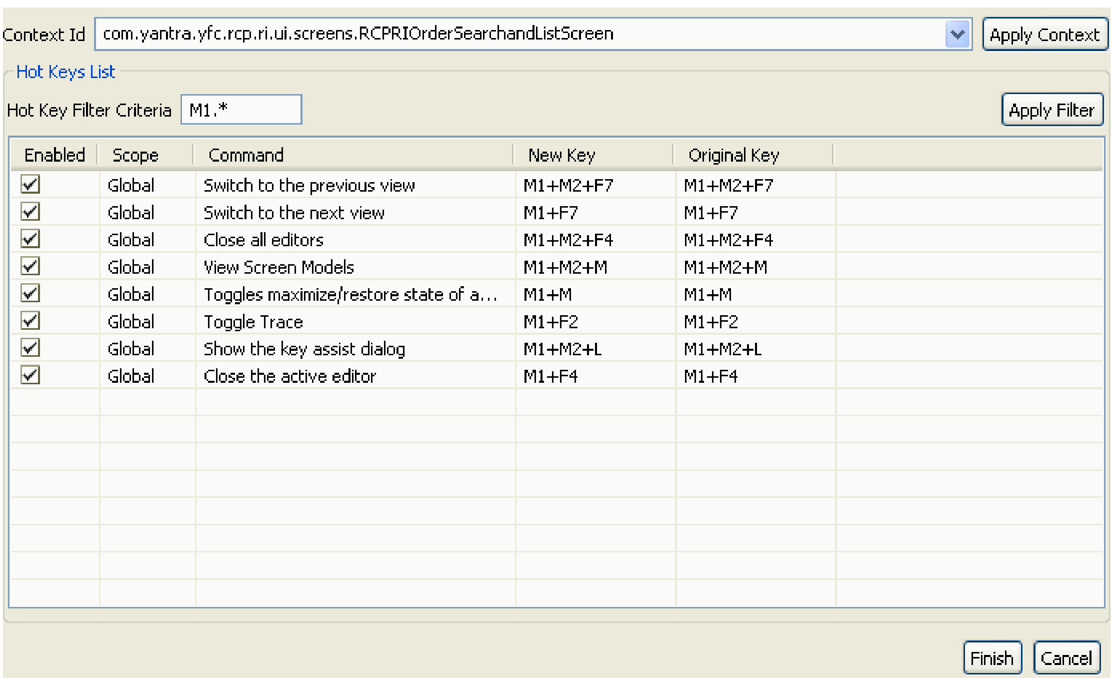


Table 4–3 Hot Key Configuration

Field	Description
Context Id	Select the context identifier for which you want to view a list of hot keys, if applicable.
Apply Context	Click this button to view all hot keys defined for the selected context identifier. A list of hot keys defined in the global context also displays.
Hot Keys List	
Hot Key Filter Criteria	Enter the custom criteria based on which you want to filter the hot keys displayed in the New Key column. You can use regular expressions for filtering the hot keys. For example, if you want to view hot keys that start with M1, specify "M1.*".
Apply Filter	Click this button to view a list of hot keys as a result of your filter criteria.
Enabled	Uncheck the box corresponding to the hot key that you want to disable. **Note:** When you uncheck a box, the complete row is grayed out.
Scope	Displays the scope or context of a particular hot key. Valid values are: "Local" and "Global". Local—indicates that the hot key is applicable for a specific screen in the application. Global—indicates that the hot key is applicable for any screen in the application.
Command	Displays the name of the command defined for a hot key.
New Key	Displays the new key sequence defined for the hot key. You can modify the new key sequence, if applicable. **Note:** If you uncheck the box, you cannot modify the new key sequence.
Original Key	Displays the original key sequence defined for the hot key. You cannot modify the original key sequence.
Finish	Click this button to save the changes made to the hot key configurations in the `plugin.xml` file of the plug-in project.

4.4 Resolving Hot Key Conflicts

Using the Rich Client Platform Extensibility Tool, you can view a list of conflicting hot keys and resolve conflicts. In the Hot Keys List panel, the specific table cells that display the hot keys with conflicts are highlighted in red.

To resolve a hot key conflict:

1. Start the Rich Client Platform Extensibility Tool.
2. Click Resolve Key Conflicts and click once on the screen. The Hot Keys window displays.

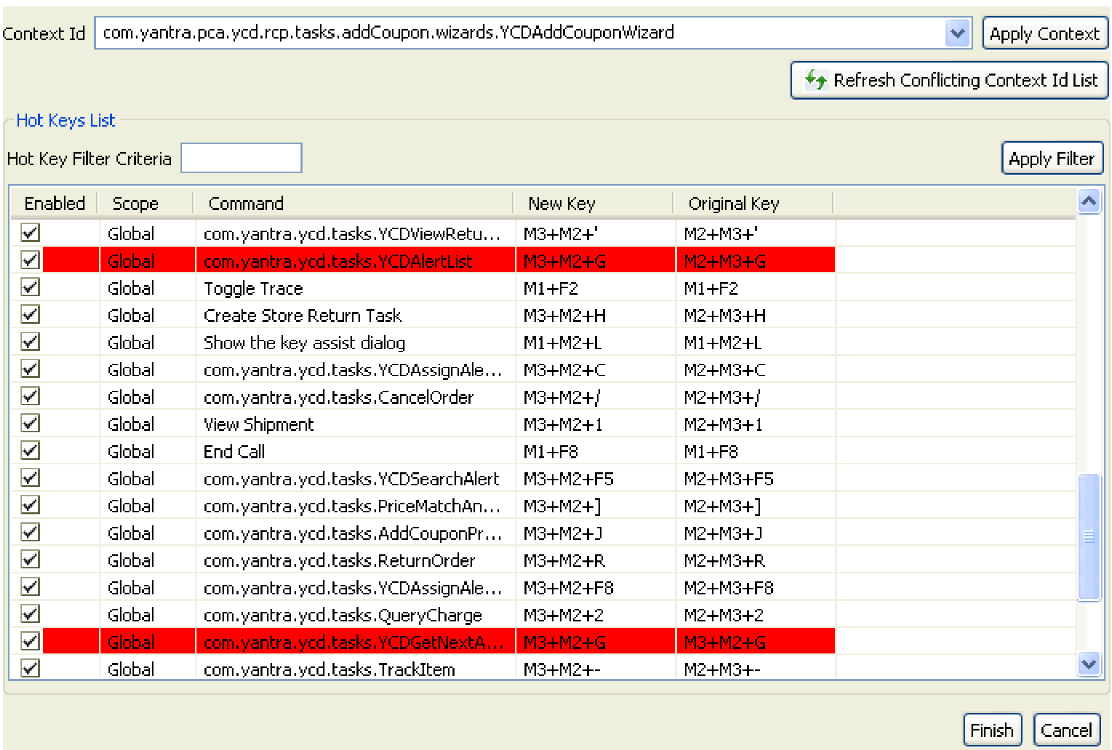


Table 4–4 Hot Key Configuration

Field	Description
Context Id	Select the context identifier for which you want to view the list of conflicting hot keys.
Apply Context	Click this button to view all hot keys defined for the selected context identifier. A list of hot keys defined in the global context also displays.
Refresh Conflicting Context Id List	Click this button to refresh the list of conflicting context identifiers.
Hot Keys List	
Hot Key Filter Criteria	Enter the custom criteria based on which you want to filter the hot keys displayed in the New Key column. You can use regular expressions for filtering the hot keys. For example, if you want to view hot keys that start with M1, specify "M1.*".
Apply Filter	Click this button to view a list of hot keys as a result of your filter criteria. The specific table cells that displays conflicted hot keys are shaded in red.
Enabled	Uncheck the appropriate box of the hot key that you want to disable. **Note:** When you uncheck any box, the complete row is grayed out.
Scope	Displays the scope or context of a particular hot key. Valid values are: • Local—indicates that the hot key is applicable for a specific screen. • Global—indicates that the hot key is applicable for any screen.
Command	Displays the name of the command defined for a hot key.
New Key	Displays the new key sequence defined for the hot key. You can modify the new key sequence, if necessary. If you uncheck the box, you cannot modify the new key sequence.

Table 4–4 Hot Key Configuration

Field	Description
Original Key	Displays the original key sequence defined for the hot key. You cannot modify the original key sequence.
Finish	Click this button to save the changes made to the hot key configurations in the `plugin.xml` file of the plug-in project.

5

Modifying New and Existing Controls in the Rich Client Platform UI

5.1 About Modifying New Controls

You can modify the newly added fields by disabling or hiding them, changing their default values, and so forth. The Rich Client Platform enables you to modify layout settings, binding attributes, and themes. You can also delete the newly added fields from the screen.

> **Note:** You cannot modify the control name of any field. However, you can set the text and combo box controls as mandatory fields.

5.2 Modifying a New Label

To modify a newly added label:

1. Start the Rich Client Platform Extensibility Tool.
2. Select a newly added label and click once. The Edit New Label window displays.

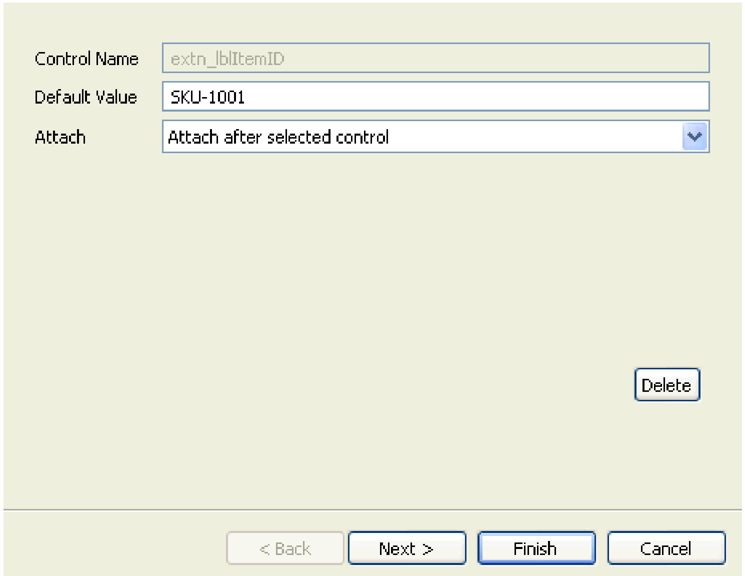


Table 5–1 Edit New Label

Field	Description
Control Name	Disabled field. The control name of any field on the form is a unique logical name used only for reference.
Default Value	Change the default value to display on the screen, if applicable.
Attach	Select the appropriate value from the drop-down list. Valid values are "Attach after selected control", "Attach before selected control", and "Attach within selected control". **Note:** If you are editing a field that is placed in a composite or group, by default, the "Attach within selected control" value displays.
Delete	Click this button to delete the newly added field from the form and extension file.

3. Click Finish. The label updates with the layout data, binding attributes, and theme.

4. Click Next to modify the layout data options.

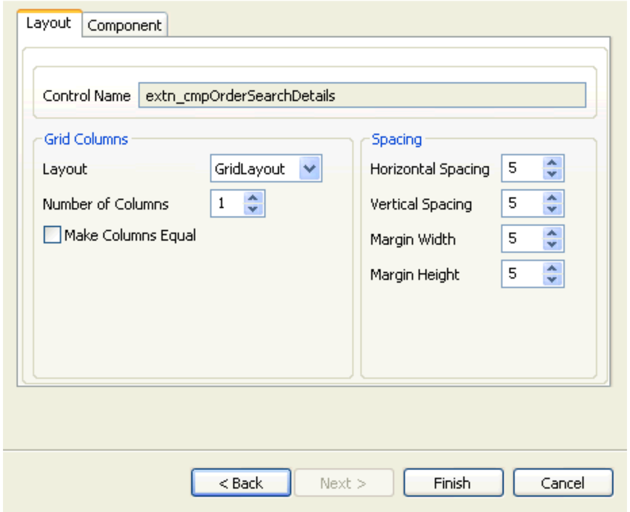


Table 5–2 Edit New Label, Layout Setup Page

Field	Description
Layout Tab This tab is used to change the layout properties for the control you want to modify. Based on the control, you can change the layout properties for the following: • Parent Composite—If you are modifying a control that is not a composite or group, the layout tab enables you to change the layout properties of the control (composite or group) to which the control belongs. You can change the position and size of the controls in the parent control. The positioning and sizing of a field depends on the layout of the parent control. • Composite or Group—If you are modifying a composite or group, the layout tab enables you to change the layout properties of the composite or group itself. **Note:** If you are modifying a control that is not a composite or group and the control name for its parent control (composite or group) is not set, the layout tab is disabled.	
Control Name	Disabled field. If you are modifying a control that is not a composite or group, this field displays the name of the parent control (composite or group) to which the control belongs. If you are modifying a composite or group, this field displays the name of the composite or group itself.
Grid Columns	
Layout	Select the layout for the control. The valid value is "GridLayout".
Number of Columns	Enter a number or click  to increase or decrease the number of columns for the layout. **Note:** This property enables you to add one or more controls to the same row by changing the number of columns in the layout.
Make Columns Equal	Check this box if you want all columns in this layout to be of equal width.
Spacing	
Horizontal Spacing	If you increase or decrease the horizontal spacing, the horizontal space between the two neighboring columns also increases or decreases.

Table 5–2 Edit New Label, Layout Setup Page

Field	Description
Vertical Spacing	If you increase or decrease the vertical spacing, the vertical space between the two neighboring columns also increases or decreases.
Margin Width	If you increase or decrease the margin width, the horizontal margin along the left and right edges of the layout also increases or decreases.
Margin Height	If you increase or decrease the margin height, the vertical margin along the top and bottom edges of the layout also increases or decreases.
Component Tab Change the properties of the layout data of the control.	
Pick Layout Like	To apply the same layout as another field on the form, select the field from the drop-down list and view its layout in the form. The drop-down list contains the control names that are siblings of the selected control.
Pick This	When you click this button, depending on the layout of the control you selected in the Pick Layout Like combo box, the values in other fields automatically displays.
Height	Enter any value between -1 and 9999 for the control.
Width	Enter any value between -1 and 9999 for the control.
Horizontal Span	If you increase the horizontal span by specifying any value between 1 and 25, the new control spans to the right of its current position.
Vertical Span	If you increase the vertical span by specifying any value between 1 and 25, the new control spans to the cell below its current position.
Grab Horizontal Space	Check this box if you want the new control to grab the extra horizontal space.
Grab Vertical Space	Check this box if you want the new control to grab the extra vertical space.

Table 5–2 Edit New Label, Layout Setup Page

Field	Description
Horizontal Alignment	Select the appropriate horizontal alignment type for the new control. Valid values are 'BEGINNING', 'CENTER', 'END', or 'FILL'. • BEGINNING aligns the new control with the left side of the horizontal space. • CENTER centers the new control within the horizontal space. • END aligns the new control with the right side of the horizontal space. • FILL fills the excess horizontal space.
Vertical Alignment	Select the appropriate vertical alignment type for the new control. Valid values are 'BEGINNING', 'CENTER', 'END', or 'FILL'. • BEGINNING aligns the new control with the top of the vertical space. • CENTER centers the new control within the vertical space. • END aligns the new control with the bottom of the vertical space. • FILL fills the excess vertical space.
Horizontal Indent	Set the horizontal indentation for the left side of the control.

5. Click Finish. The label is updated with the binding attributes and theme.
6. Click Next to modify the binding options. For field value descriptions, see Table 3–3.
7. Click Finish. The label is updated with the theme binding.
8. Click Next to modify the theme binding.

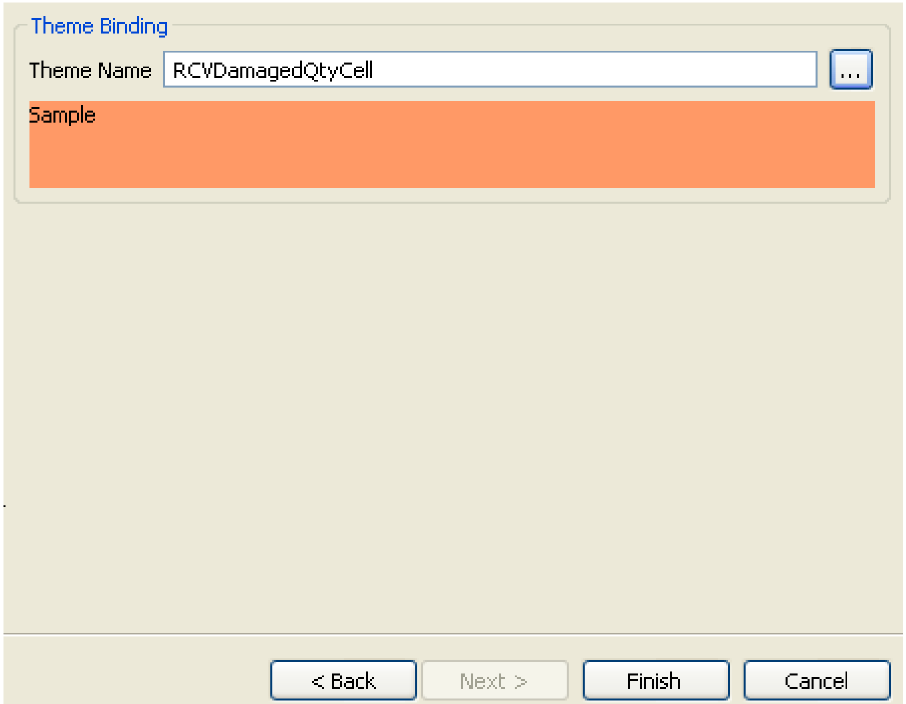


Table 5–3 Edit New Label, Theme Binding Page

Field	Description
Theme Name	Enter the theme name you want to apply for the field, if applicable. Click ... to view the available themes. The Themes Tree pop-up window displays. Select the appropriate theme name you want to apply. You can also press Ctrl+Space and select the name of the theme you want to apply from the drop-down list.

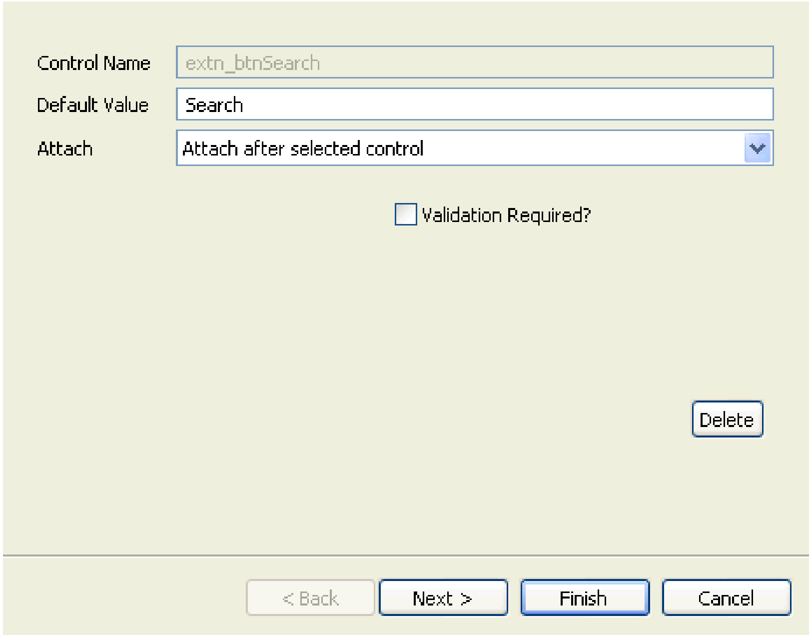
9. Click Finish.
10. Click to save the changes made to the extension file.
11. If you have specified the default value, bindings, or theme entry for the checkbox, you must synchronize the resource files. For more

information about synchronizing resource files, see Chapter 6, "Synchronizing New and Existing UI Resources".

5.3 Modifying a New Button

To modify a newly added button:

1. Start the Rich Client Platform Extensibility Tool.
2. Select the newly added button and click once. The Edit New Button window displays.



Note: If you attempt to modify a button that is placed in the row layout container, the Layout Setup Page window is disabled.

Table 5–4 Edit New Button

Field	Description
Control Name	Disabled field. The control name of any field on the form is a unique logical name used only for reference.
Default Value	Change the default value to display on the screen, if applicable.
Attach	Select the appropriate value from the drop-down list. Valid values are "Attach after selected control", "Attach before selected control", and "Attach within selected control". **Note:** If you are editing a field that is placed in a composite or group, by default, the "Attach within selected control" value displays.
Validation Required?	Check this box to validate a new field, if applicable. The Rich Client Platform invokes the validation method on the extended behavior class of this form when the field loses focus.
Delete	Click this button to delete the newly added field from the form and extension file.

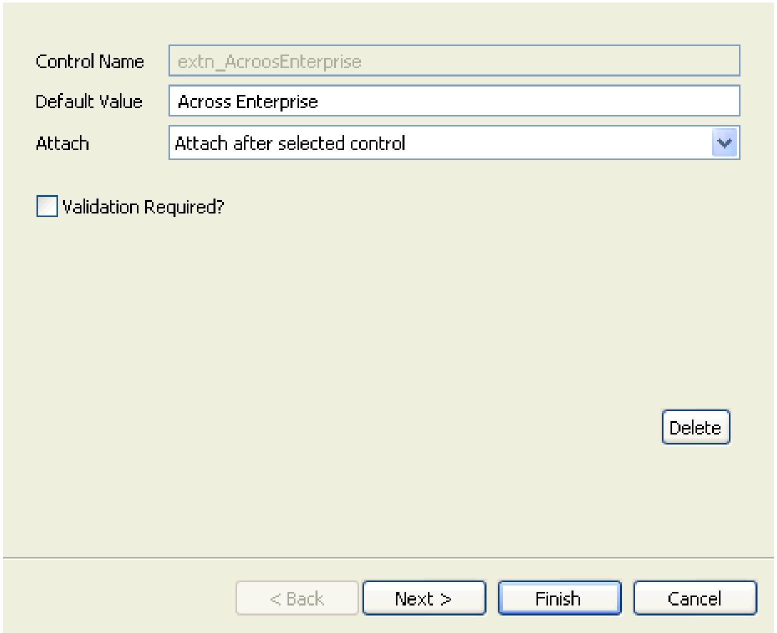
3. Click Finish. The button is updated with the layout data.
4. Click Next to modify the layout data options. For field value descriptions, see Table 5–2.
5. Click Finish. The button is updated with the theme binding.
6. Click Next to modify the theme binding. For field value descriptions, see Table 5–3.
7. Click Finish.
8. Click to save the changes made to the extension file.
9. If you have specified the default value, bindings, or theme entry for the button, you must synchronize the resource files. For more information about synchronizing resource files, see Chapter 6, "Synchronizing New and Existing UI Resources".

5.4 Modifying a New Checkbox

To modify a newly added checkbox:

1. Start the Rich Client Platform Extensibility Tool.
2. Select the newly added checkbox and click once.

   The Edit New Checkbox window displays. For field value descriptions, see Table 5–4.



Note: If you attempt to modify a checkbox that is placed in the row layout container, the Layout Setup Page window disabled.

3. Click Finish. The checkbox is updated with layout data, binding attributes, and theme binding.
4. Click Next to modify the layout data options. For field value descriptions, see Table 5–2.

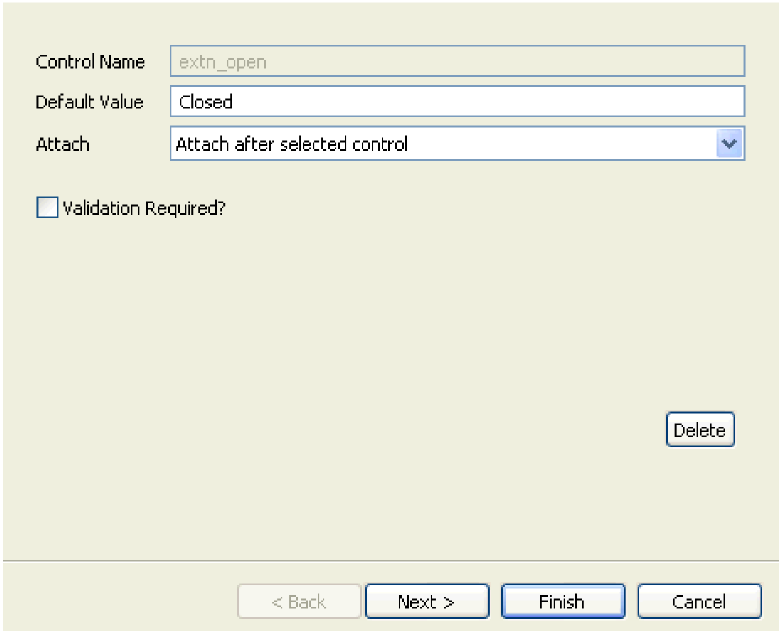
5. Click Finish. The checkbox is updated with the binding attributes and theme binding.
6. Click Next to modify the binding options. For field value descriptions, see Table 3–7.
7. Click Finish. The checkbox is updated with the theme binding.
8. Click Next to modify the theme binding. For field value descriptions, see Table 5–3.
9. Click Finish.
10. Click  to save the changes made to the extension file.
11. If you have specified the default value, bindings, or theme entry for the checkbox, you must synchronize the resource files. For more information about synchronizing resource files, see Chapter 6, "Synchronizing New and Existing UI Resources".

5.5 Modifying a New Radio Button

To modify a newly added radio button:

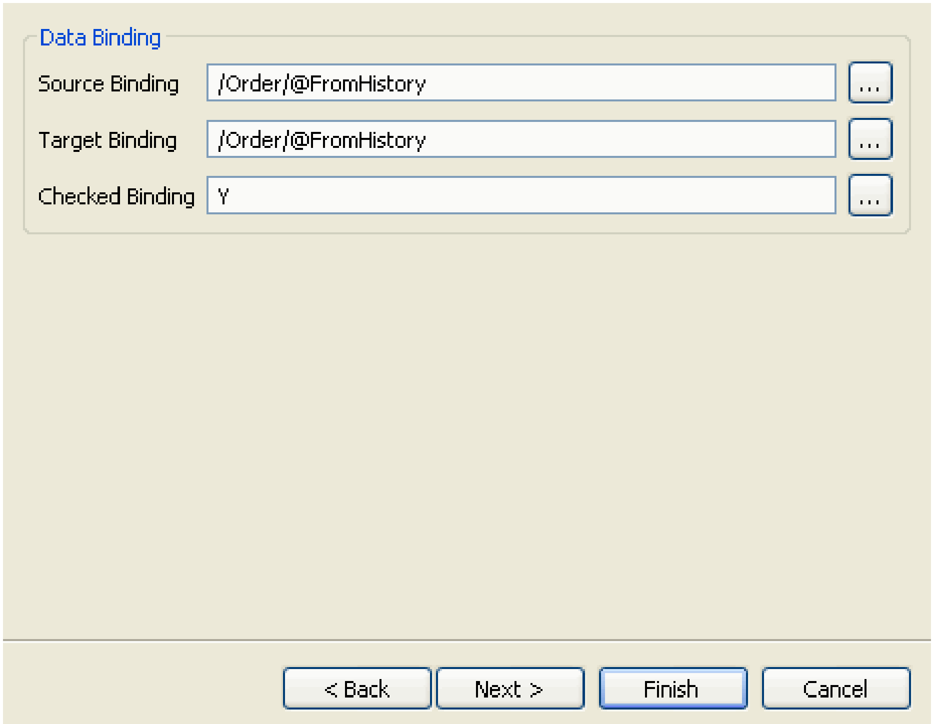
1. Start the Rich Client Platform Extensibility Tool.
2. Select a newly added radio button and click once.

   The Edit New Radio Button window displays. For field value descriptions, see Table 5–4.



Note: If you attempt to modify a radio button that is placed in the row layout container, the Layout Setup Page window is disabled.

3. Click Finish. The radio button is updated with the layout data, binding attributes, and theme binding.
4. Click Next to modify the layout data options. For field value descriptions, see Table 5–2.
5. Click Finish. The radio button is updated with the binding attributes and theme binding.
6. Click Next to modify the binding options. For field value descriptions, see Table 3–7.



7. Click Finish. The newly added radio button is updated with the existing theme.
8. Click Next to modify the theme binding. For field value descriptions, see Table 5–3.
9. Click Finish.
10. Click  to save the changes made to the extension file.
11. If you have specified the default value, bindings, or theme entry for the radio button, you must synchronize the resource files. For more information about synchronizing resource files, see Chapter 6, "Synchronizing New and Existing UI Resources".

5.6 Modifying a New Text Box

To modify a newly added text box:

1. Start the Rich Client Platform Extensibility Tool.
2. Select a newly added text box and click once. The Edit New Text Box window displays.

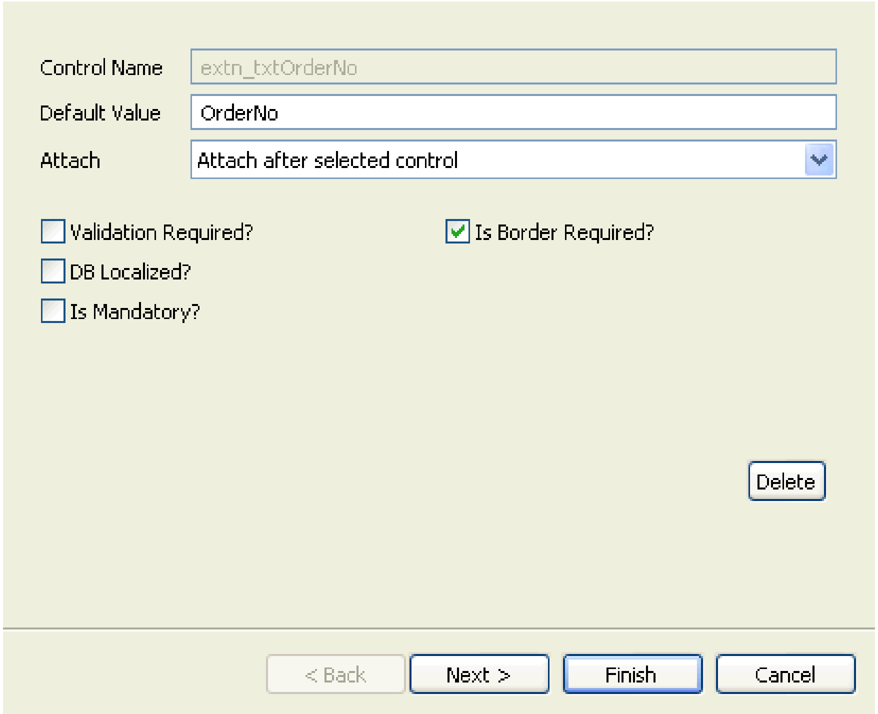
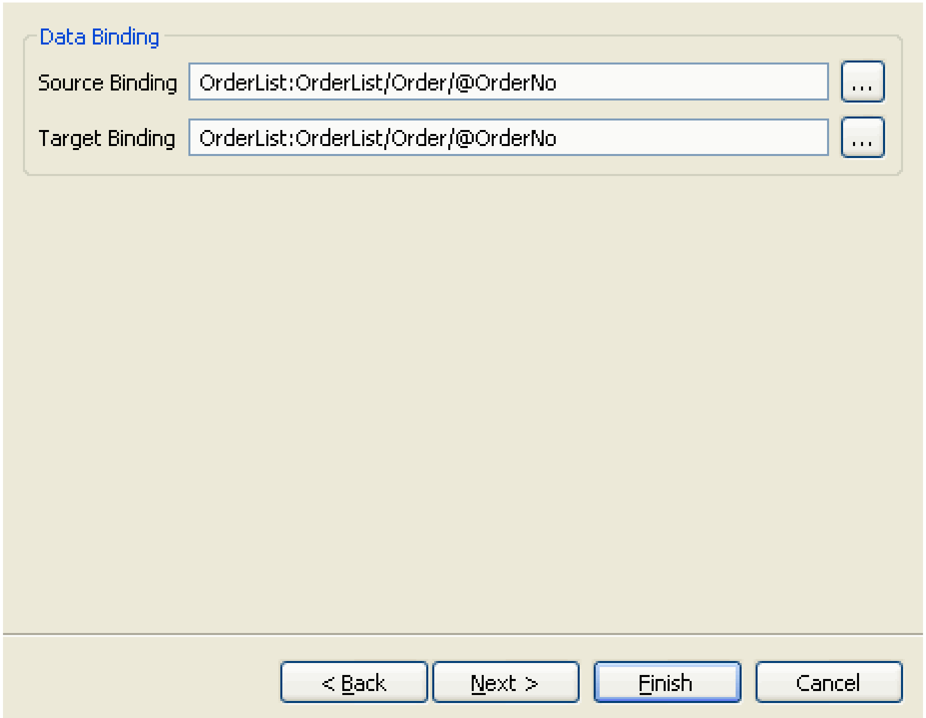


Table 5–5 Edit New Text Box

Field	Description
Control Name	Disabled field. The control name of any field on the form is a unique logical name used only for reference.
Default Value	Change the default value to display on the screen, if applicable.
Attach	Select the appropriate value from the drop-down list. Valid values are "Attach after selected control", "Attach before selected control", and "Attach within selected control". **Note:** If you are editing a field that is placed in a composite or group, by default, the "Attach within selected control" value displays.
Validation Required?	Check this box to validate a new field, if applicable. The Rich Client Platform invokes the validation method on the extended behavior class of this form when the field loses focus.
DB Localized?	Check this box to retrieve data for the field from a localized database (DB) column, if applicable.
Is Mandatory?	Check this box if you want to make the field mandatory.
Is Border Required?	Uncheck this box if you want to remove the border around the field.
Delete	Click this button to delete the newly added field from the form and extension file.

3. Click Finish. The text box is updated with the layout data, binding attributes, and theme binding.
4. Click Next to modify the layout data options. For field value descriptions, see Table 5–2.
5. Click Finish. The text box is updated with the binding attributes and theme binding.
6. Click Next to modify the binding options. For field value descriptions, see Table 3–7.

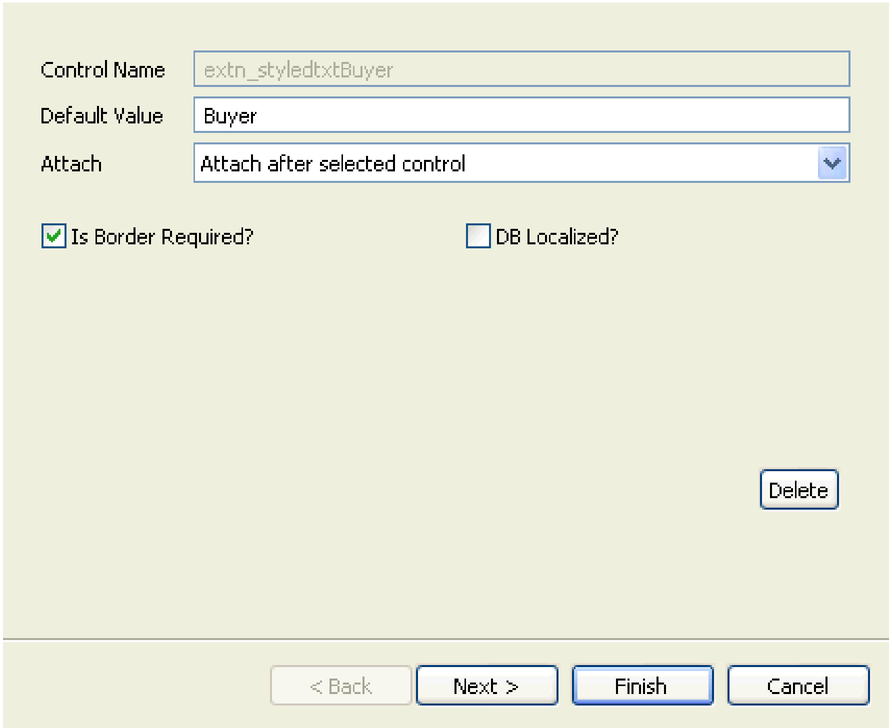


7. Click Finish. The text box is edited with the theme binding.
8. Click Next to modify the theme binding. For field value descriptions, see Table 5–3.
9. Click Finish.
10. Click  to save the changes made to the extension file.
11. If you have specified the default value, bindings, or theme entry for the text box, you must synchronize the resource files. For more information about synchronizing resource files, see Chapter 6, "Synchronizing New and Existing UI Resources".

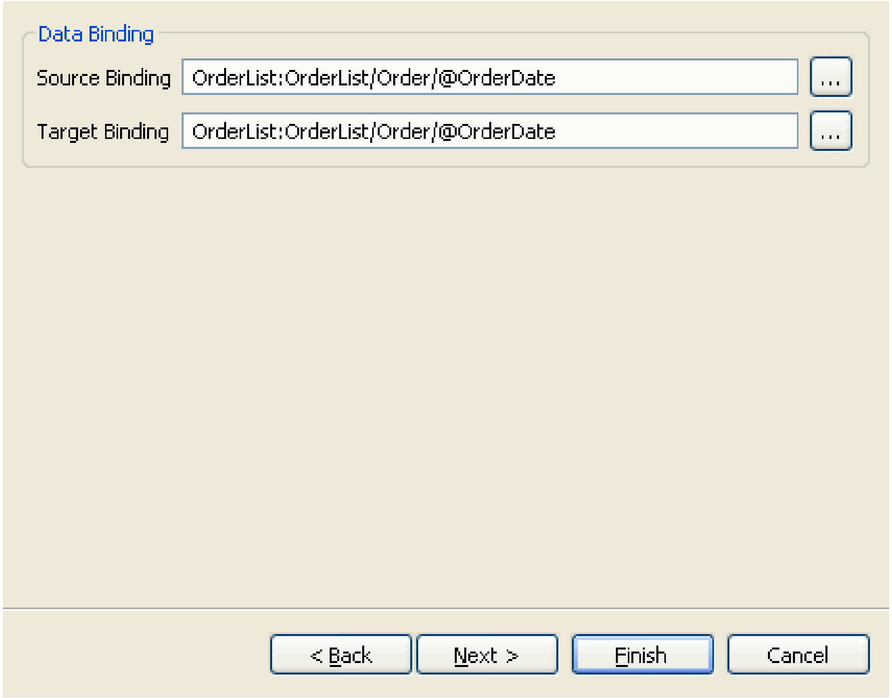
5.7 Modifying New StylesText Component

To modify a newly added styled text:

1. Start the Rich Client Platform Extensibility Tool.
2. Select a newly added styled text and click once. The Edit New Styled Text window displays. For field value descriptions, see Table 5–5.



3. Click Finish. The styled text is updated with the layout data, binding attributes, and theme binding.
4. Click Next to modify the layout data options. For field value descriptions, see Table 5–2.
5. Click Finish. The styled text is updated with the binding attributes and theme binding.
6. Click Next to modify the binding options. For field value descriptions, see Table 3–7.



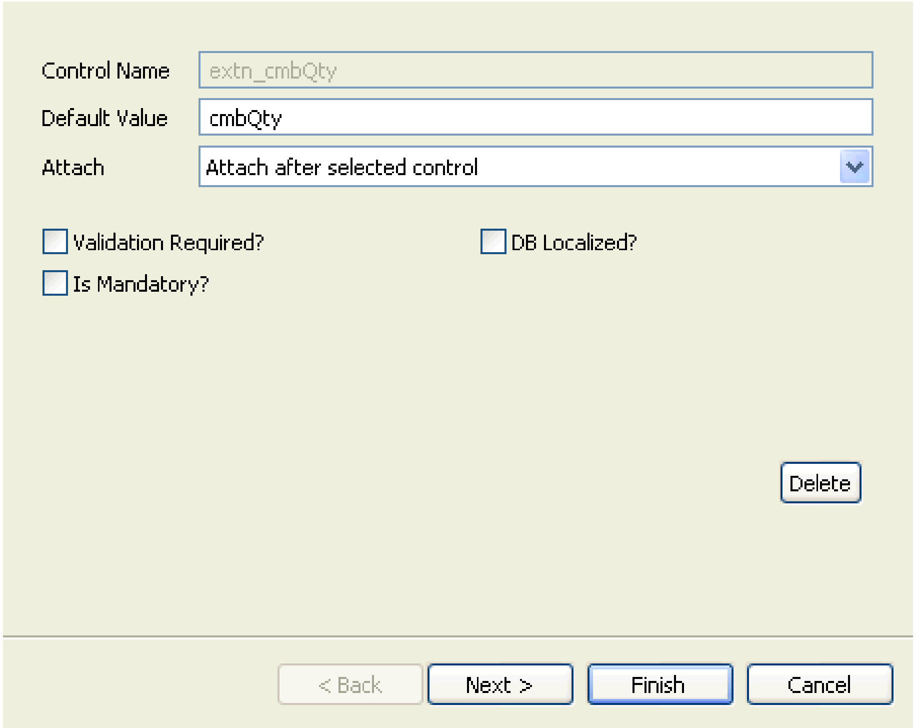
7. Click Finish. The styled text is updated with the theme binding.
8. Click Next to modify the theme binding. For field value descriptions, see Table 5–3.
9. Click Finish.
10. Click to save the changes made to the extension file.
11. If you have specified the default value, bindings, or theme entry for the styled text, you must synchronize the resource files. For more information about synchronizing resource files, see Chapter 6, "Synchronizing New and Existing UI Resources".

5.8 Modifying a New Combo Box

To modify a newly added combo box:

1. Start the Rich Client Platform Extensibility Tool.

2. Select a newly added combo box and click once. The Edit New Combo Box window displays. For field value descriptions, see Table 5–5.



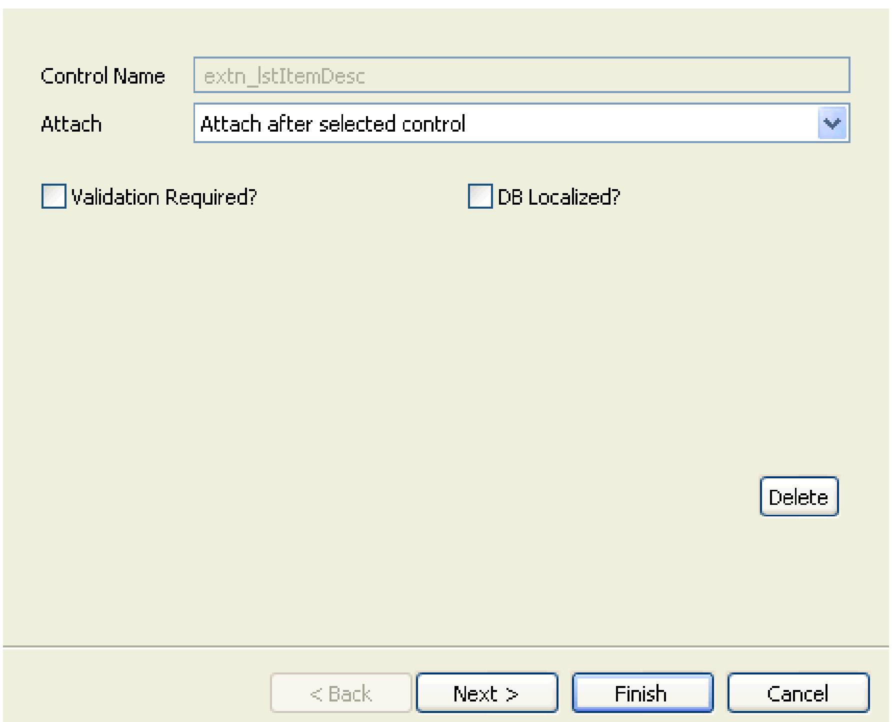
3. Click Finish. The combo box is updated with the layout data, binding attributes, and theme binding.
4. Click Next to modify the layout data options. For field value descriptions, see Table 5–2.
5. Click Finish. The combo box is updated with the binding attributes and theme binding.
6. Click Next to modify the binding options. For field value descriptions, see Table 3–9.
7. Click Finish. The combo box is updated with the theme binding.
8. Click Next to modify the theme binding. For field value descriptions, see Table 5–3.

9. Click Finish.
10. Click  to save the changes made to the extension file.
11. If you have specified the default value, bindings, or theme entry for the combo box, you must synchronize the resource files. For more information about synchronizing resource files, see Chapter 6, "Synchronizing New and Existing UI Resources".

5.9 Modifying a New List Box

To modify a newly added list box:

1. Start the Rich Client Platform Extensibility Tool.
2. Select a newly added list box and click once. The Edit New List Box window displays. For field value descriptions, see Table 5–5.



3. Click Finish. The list box is updated with the layout data, binding attributes, and theme binding.
4. Click Next to modify the layout data options. For field value descriptions, see Table 5–2.
5. Click Finish. The list box is updated with the binding attributes and theme binding.
6. Click Next to modify the binding options. For field value descriptions, see Table 3–9.
7. Click Finish. The list box is updated with the theme binding.
8. Click Next to modify the theme binding. For field value descriptions, see Table 5–3.
9. Click Finish.
10. Click  to save the changes made to the extension file.
11. If you have specified the default value, bindings, or theme entry for the list box, you must synchronize the resource files. For more information about synchronizing resource files, see Chapter 6, "Synchronizing New and Existing UI Resources".

5.10 Modifying a New Table Column

To modify a newly added table column:

1. Start the Rich Client Platform Extensibility Tool.
2. Select a newly added column in a table and click once. The Edit New Table Column window displays.

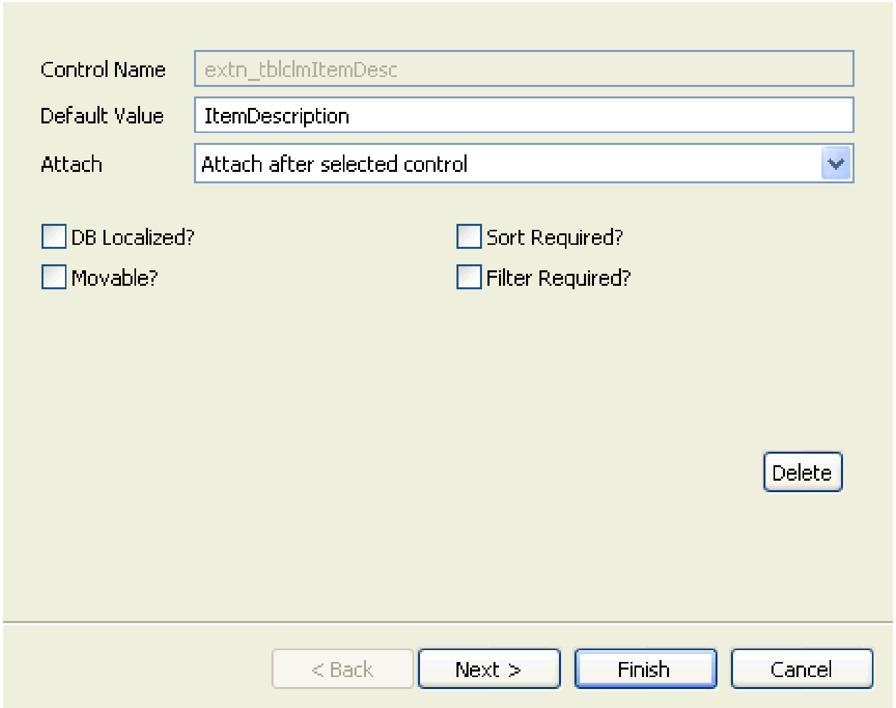


Table 5–6 Edit New Table Column

Field	Description
Control Name	Disabled field. Control Name of any field on the form is a unique logical name used only for reference.
Default Value	Change the default value to display on the screen, if applicable.
Attach	Select the appropriate value from the drop-down list. Valid values are "Attach after selected control" and "Attach before selected control".
DB Localized?	Check this box to retrieve data for the field from a localized database (DB) column, if applicable.
Movable?	Check this box to move table columns, if applicable.

Table 5–6 Edit New Table Column

Field	Description
Sort Required?	Check this box to sort the data in a particular order, if applicable.
Filter Required?	Check this box to filter the data, if applicable.
Delete	Click this button to delete the newly added field from the form and extension file.

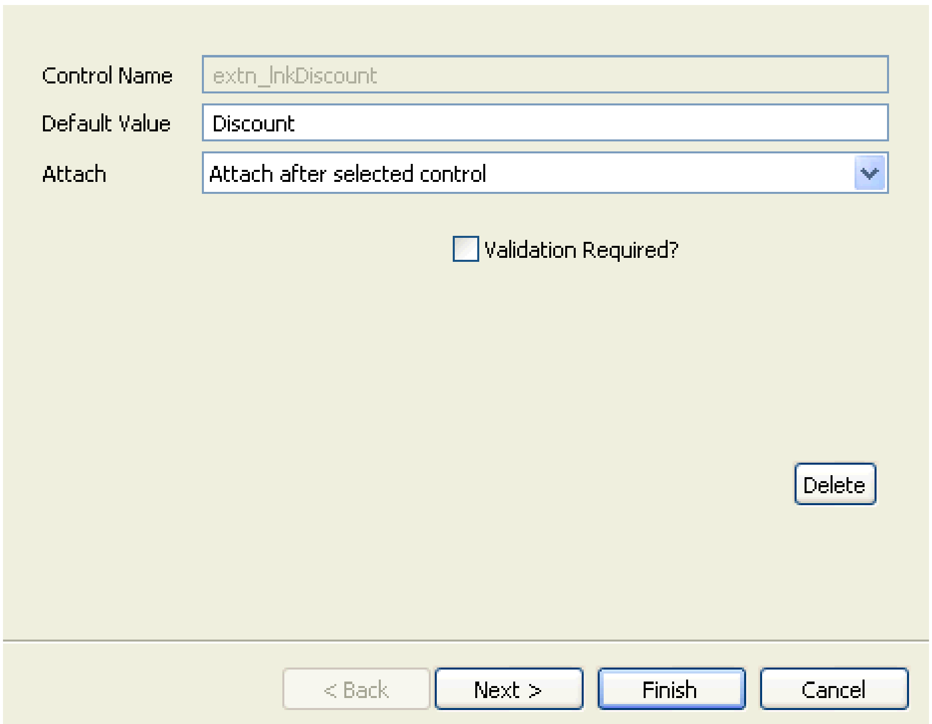
3. Click Finish to add the new column with the default column layout data, theme, and null binding attributes.
4. Click Next to specify the column layout options. For field value descriptions, see Table 3–11.
5. Click Finish. The table column is updated with the binding attributes and theme binding.
6. Click Next to modify the binding options. For field value descriptions, see Table 3–12.
7. Click Finish. The table column is updated with the theme binding.
8. Click Next to modify the theme binding. For field value descriptions, see Table 5–3.
9. Click Finish.
10. Click  to save the changes made to the extension file.
11. If you have specified the default value, bindings, or theme entry for the table column, you must synchronize the resource files. For more information about synchronizing resource files, see Section 6, "Synchronizing New and Existing UI Resources".

5.11 Modifying a New Link

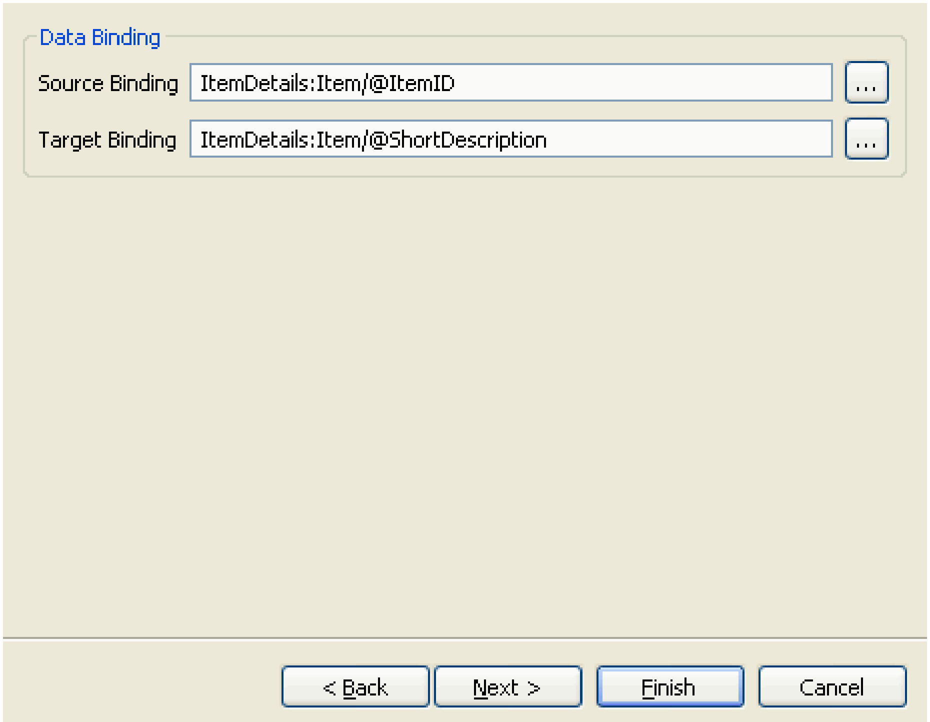
To modify a newly added link:

1. Start the Rich Client Platform Extensibility Tool.
2. Select a newly added link and click once. The Edit New Link window displays. For field value descriptions, see Table 5–5.

Note: If you attempt to modify a link that is placed in the row layout container, the Layout Setup Page window disabled.



3. Click Finish. The link is updated with the layout data, binding attributes, and theme binding.
4. Click Next to modify the layout data options. For field value descriptions, see Table 5–2.
5. Click Finish. The link is updated with the binding attributes and theme binding.
6. Click Next to modify the binding options. For field value descriptions, see Table 3–9.



7. Click Finish. The link is updated with the theme binding.
8. Click Next to modify the theme binding. For field value descriptions, see Table 5–3.
9. Click Finish.
10. Click  to save the changes made to the extension file.
11. If you have specified the default value, bindings, or theme entry for the link, you must synchronize the resource files. For more information about synchronizing resource files, see Chapter 6, "Synchronizing New and Existing UI Resources".

5.12 Modifying a New Composite

To modify a newly added composite:

1. Start the Rich Client Platform Extensibility Tool.
2. Select a newly added composite and click once. The Edit New Composite window displays.

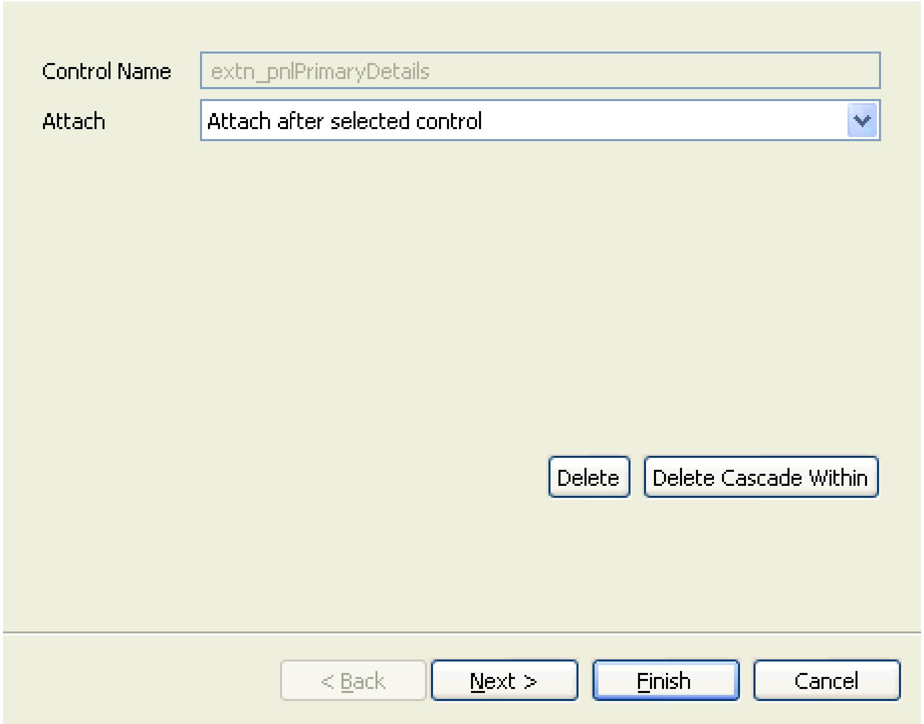


Table 5–7 Edit New Composite

Field	Description
Control Name	Disabled field. The control name of any field on the form is a unique logical name used only for reference.
Attach	Select the appropriate value from the drop-down list. Valid values are "Attach after selected control", "Attach before selected control", and "Attach within selected control". **Note:** If you are editing a field that is placed in a composite or group, by default, the "Attach within selected control" value displays.
Delete	Click this button to delete the field from the form and extension file.
Delete Cascade Within	Click this button to delete the field and its sibling controls from the form and extension file.

3. Click Finish. The newly added composite is edited with the existing layout data.
4. Click Next to modify the layout data options. For field value descriptions, see Table 5–2.
5. Click Finish.
6. Click to save the changes made to the extension file.

5.13 Modifying a New Group

To modify a newly added group:

1. Start the Rich Client Platform Extensibility Tool.
2. Select a newly added group and click once. The Edit New Group window displays.

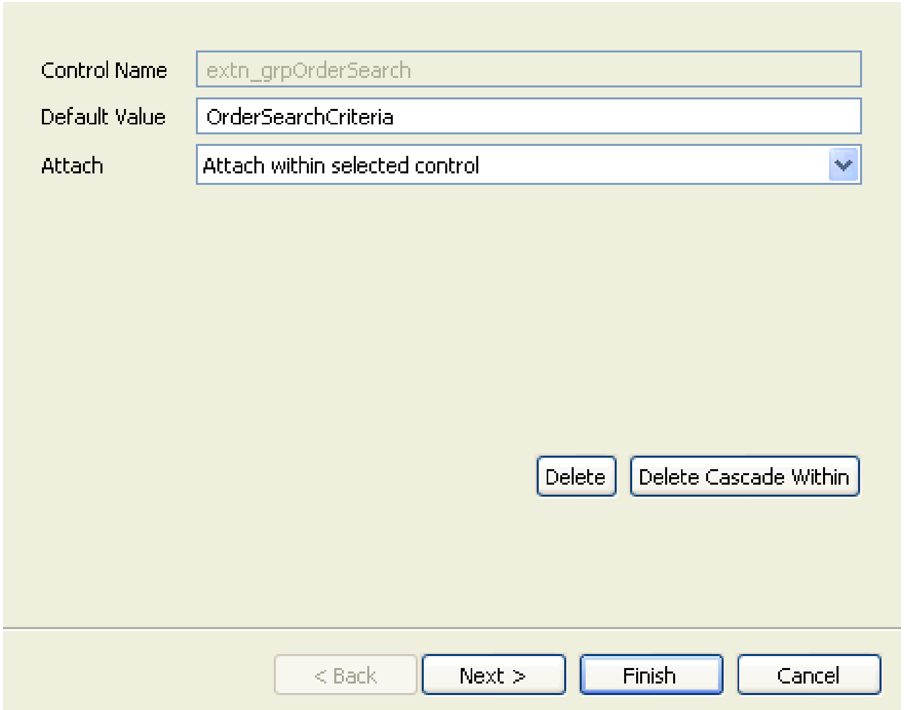


Table 5–8 Edit New Group

Field	Description
Control Name?	Disabled field. The control name of any field on the form is a unique logical name used only for reference.
Default Value	Change the default value to display on the screen, if applicable.
Attach	Select the appropriate value from the drop-down list. Valid values are "Attach after selected control", "Attach before selected control", and "Attach within selected control". **Note:** If you are editing a field that is placed in a composite or group, by default, the "Attach within selected control" value displays.

Table 5–8 Edit New Group

Field	Description
Delete	Click this button to delete the newly added field from the form and extension file.
Delete Cascade Within	Click this button to delete the newly added field and its sibling controls from the form and extension file.

3. Click Finish. The group is updated with the layout data.
4. Click Next to modify the layout data options. For field value descriptions, see Table 5–2.
5. Click Finish.
6. Click to save the changes made to the extension file.
7. If you have specified the default value, bindings, or theme entry for the group, you must synchronize the resource files. For more information about synchronizing resource files, see Chapter 6, "Synchronizing New and Existing UI Resources".

5.14 About Modifying Existing Controls

You can modify the fields provided with the Rich Client Platform application by disabling them, hiding them, changing their default values, and so forth. The Rich Client Platform provides the ability to modify layout settings and theme binding. You can also delete the existing fields from the screen.

> **Note:** You cannot modify the control name of any existing fields.

> **Note:** Provided fields that are mandatory cannot be made optional.

5.15 Modifying an Existing Label, Composite, or Group

To modify an existing label, composite, or group:

1. Start the Rich Client Platform Extensibility Tool.
2. Click to show hidden and disabled fields, if applicable.
3. Select an existing field and click once. The Edit Existing Field window displays.

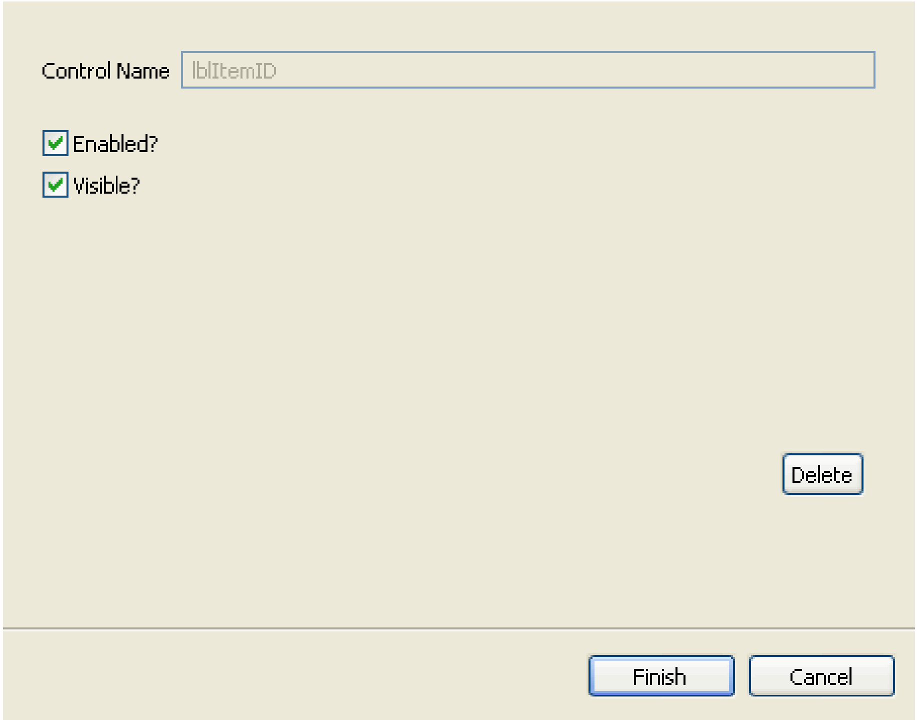


Table 5–9 Edit Existing Field

Field	Description
Control Name	Disabled field. The control name of any field on the form is a unique logical name used only for reference.
Enabled?	Uncheck this box if you want to disable the field.
Visible?	Uncheck this box if you want to hide the field. **Note:** You cannot hide a composite or group if it contains mandatory fields.
Delete	This button displays if you attempt to modify an existing field that is already modified. When you click this button, the field is deleted from the extension file. Additionally, all the modifications made to this field are deleted. The default settings for this field are automatically restored.

4. Click Next to modify the layout data options. For field value descriptions, see Table 5–2.
5. Click Next to modify the theme binding. For field value descriptions, see Table 5–3.
6. Click Finish.
7. Click to save the changes made to the extension file.
8. If you have specified the theme entry for the field, you must synchronize the resource files. For more information about synchronizing resource files, see Chapter 6, "Synchronizing New and Existing UI Resources".

5.16 Modifying an Existing Button, Checkbox, Radio Button, List Box, or Link

To modify an existing button, checkbox, radio button, list box, or link:

1. Start the Rich Client Platform Extensibility Tool.
2. Click to show hidden and disabled fields, if applicable.
3. Select an existing field and click once. The Edit Existing Field window displays.

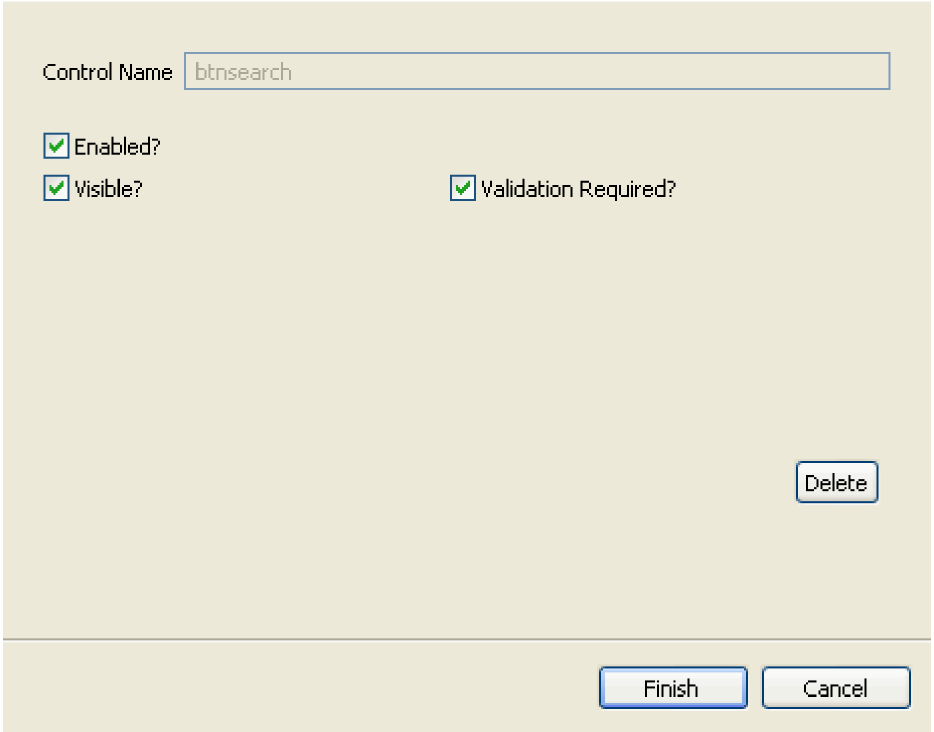


Table 5–10 Edit Existing Field

Field	Description
Control Name	Disabled field. The control name of any field on the form is a unique logical name used only for reference.
Enabled?	Uncheck this box if you want to disable the field.
Visible?	Uncheck this box if you want to hide the field.

Table 5–10 Edit Existing Field

Field	Description
Validation Required?	Check this box to validate a field, if applicable. The Rich Client Platform invokes the validation method on the extended behavior class of this form when the field loses focus.
Delete	This button displays if you attempt to modify an existing field that is already modified. When you click this button, the field is deleted from the extension file. Additionally, all modifications made to this field are deleted. The default settings for this field are automatically restored.

4. Click Next to modify the layout data options. For field value descriptions, see Table 5–2.
5. Click Next to modify the theme binding. For field value descriptions, see Table 5–3.
6. Click Finish.
7. Click to save the changes made to the extension file.
8. If you have specified the default theme entry for the field, you must synchronize the resource files. For more information about synchronizing resource files, see Chapter 6, "Synchronizing New and Existing UI Resources".

5.17 Modifying an Existing Text Box

To modify an existing text box:

1. Start the Rich Client Platform Extensibility Tool.
2. Click to show hidden and disabled fields, if applicable.
3. Select an existing field and click once. The Edit Existing Field window displays.

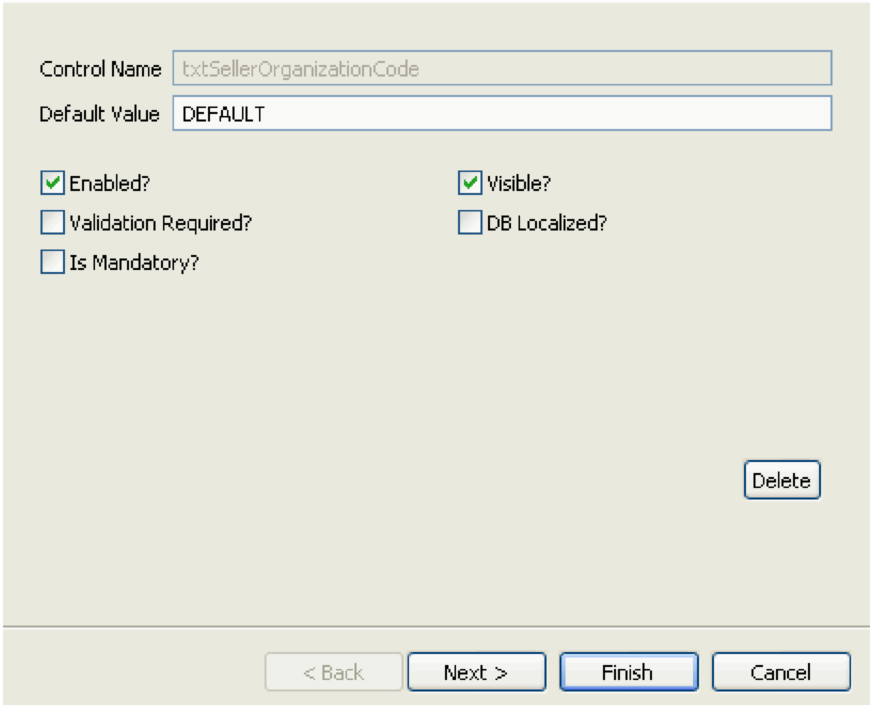


Table 5–11 Edit Existing Field

Field	Description
Control Name	Disabled field. The control name of any field on the form is a unique logical name used only for reference.
Default Value	Change the default value to display on the screen, if applicable.
Enabled?	Uncheck this box if you want to disable the field.
Validation Required?	Check this box to validate a field, if applicable. The Rich Client Platform invokes the validation method on the extended behavior class of this form when the field loses focus.
Is Mandatory?	Check this box if you want make this field mandatory.
Visible?	Uncheck this box if you want to hide the field.

Table 5–11 Edit Existing Field

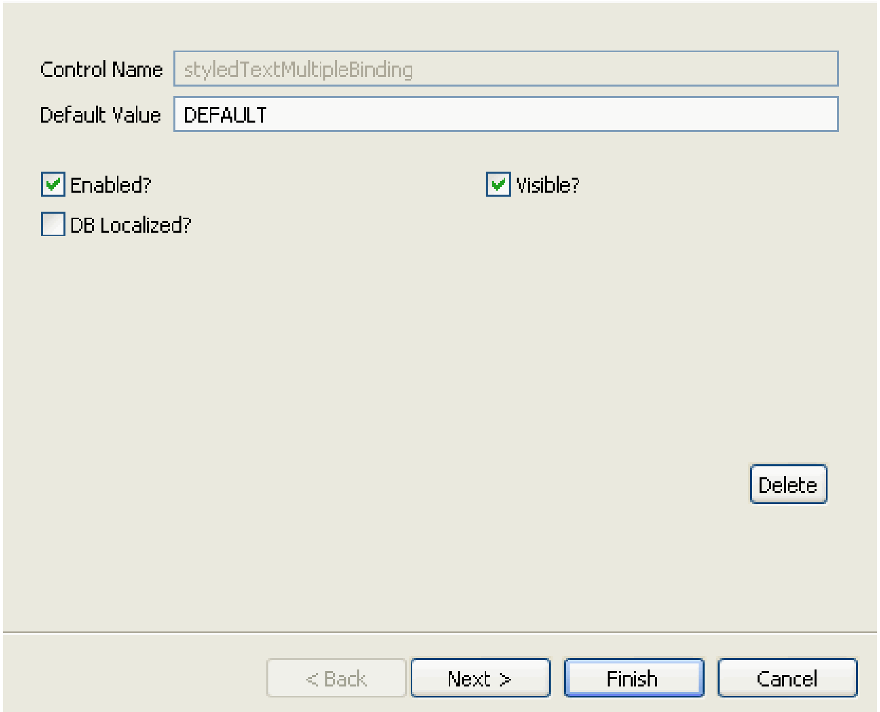
Field	Description
DB Localized?	Check this box to retrieve data for the field from a localized database (DB) column, if applicable.
Delete	This button displays if you attempt to modify an existing field that is already modified. When you click this button, the field is deleted from the extension file. Additionally, all modifications made to this field are deleted. The default settings for this field are automatically restored.

4. Click Next to modify the layout data options. For field value descriptions, see Table 5–2.
5. Click Next to modify the theme binding. For field value descriptions, see Table 5–3.
6. Click Finish.
7. Click to save the changes made to the extension file.
8. If you have specified the default value, bindings, or theme entry for the field, you must synchronize the resource files. For more information about synchronizing resource files, see Chapter 6, "Synchronizing New and Existing UI Resources".

5.18 Modifying an Existing StyledText Component

To modify an existing StyledText component:

1. Start the Rich Client Platform Extensibility Tool.
2. Click to show hidden and disabled fields, if applicable.
3. Select an existing field and click once. The Edit Existing Field window displays. For field value descriptions, see Table 5–5.



4. Click Next to modify the layout data options. For field value descriptions, see Table 5–2.
5. Click Next to modify the theme binding. For field value descriptions, see Table 5–3.
6. Click Finish.
7. Click  to save the changes made to the extension file.
8. If you have specified the default value, bindings, or theme entry for the field, you must synchronize the resource files. For more information about synchronizing resource files, see Appendix 6, "Synchronizing New and Existing UI Resources".

5.19 Modifying an Existing Table Column

To modify an existing table column:

1. Start the Rich Client Platform Extensibility Tool.
2. Click to show hidden and disabled fields, if applicable.
3. Select an existing column in the table and click once. The Edit Existing Field window displays.

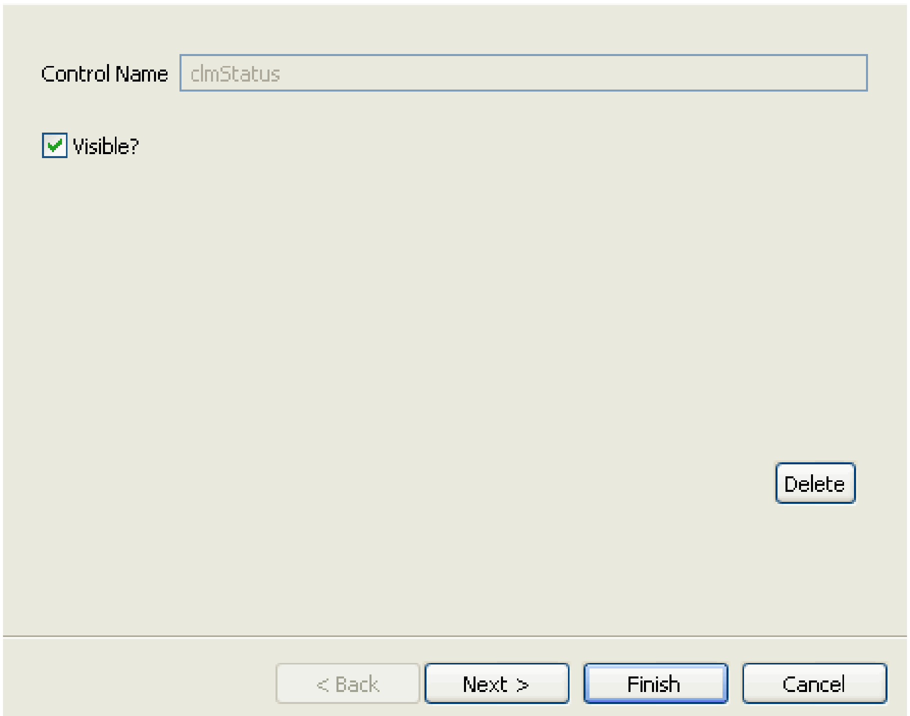


Table 5–12 Edit Existing Field

Field	Description
Control Name	Disabled field. The control name of any field on the form is a unique logical name used only for reference.

Table 5–12 Edit Existing Field

Field	Description
Visible?	Uncheck this box if you want to hide the table column.
Delete	This button displays if you attempt to modify an existing field that is already modified. When you click this button, the field is deleted from the extension file. Also, all modifications made to this field is deleted, and the default settings for this field are automatically restored.

4. Click Next to modify the column layout options. For field value descriptions, see Table 3–11.
5. Click Next to modify the theme binding. For field value descriptions, see Table 3–13.
6. Click Finish.
7. Click to save the changes made to the extension file.
8. If you have specified the theme entry for the table column, you must synchronize the resource files. For more information about synchronizing resource files, see Appendix 6, "Synchronizing New and Existing UI Resources".

5.20 Modifying an Existing Combo Box

To modify an existing combo box:

1. Start the Rich Client Platform Extensibility Tool.
2. Click to show hidden and disabled fields, if applicable.
3. Select an existing field and click once. The Edit Existing Field window displays.

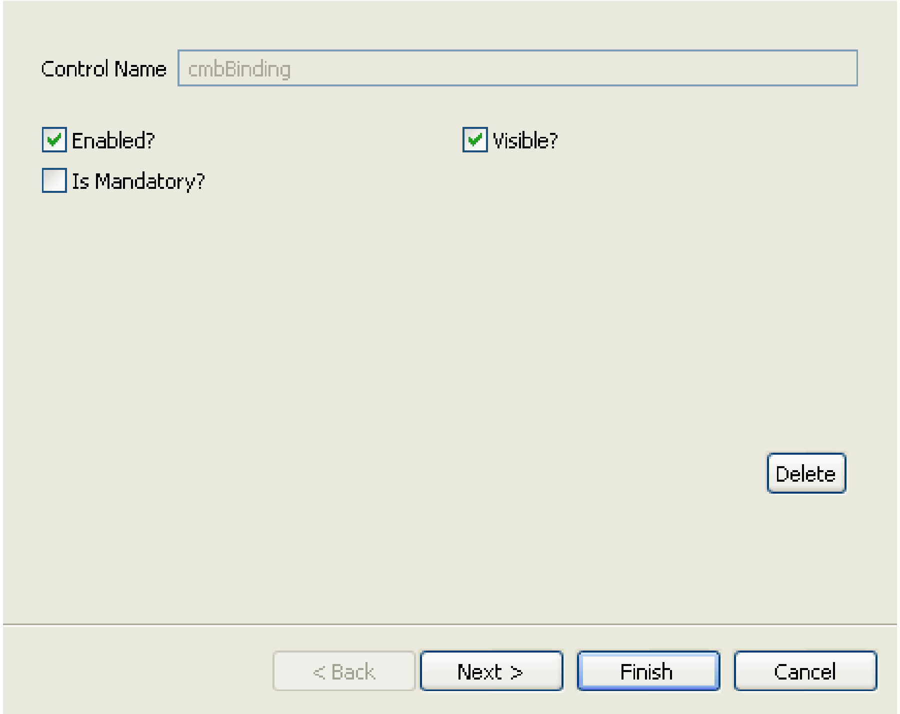


Table 5–13   Edit Existing Field

Field	Description
Control Name	Disabled field. The control name of any field on the form is a unique logical name used only for reference.
Enabled?	Uncheck this box if you want to disable the field.
Visible?	Uncheck this box if you want to hide the field.

Table 5–13 Edit Existing Field

Field	Description
Is Mandatory?	Check this box if you want to make the field mandatory.
Delete	This button displays only if you attempt to modify fields that are already modified. When you click this button, the field is deleted from the extension file. Additionally, all modifications made to this field are deleted. The default settings for this field are automatically restored.

4. Click Next to modify the layout data options. For field value descriptions, see Table 5–2.
5. Click Next to modify the theme binding. For field value descriptions, see Table 5–3.
6. Click Finish.
7. Click  to save the changes made to the extension file.
8. If you have specified the theme entry for the field, you must synchronize the resource files. For more information about synchronizing resource files, see Chapter 6, "Synchronizing New and Existing UI Resources".

6

Synchronizing New and Existing UI Resources

6.1 Synchronizing Bundle Entries

Whenever you add new fields, you must synchronize the existing bundle file with the new bundle entries.

To synchronize the bundle file:

1. Start the Rich Client Platform Extensibility Tool.
2. Click . The Synchronize pop-up window displays. Select the Bundle Entries tab.

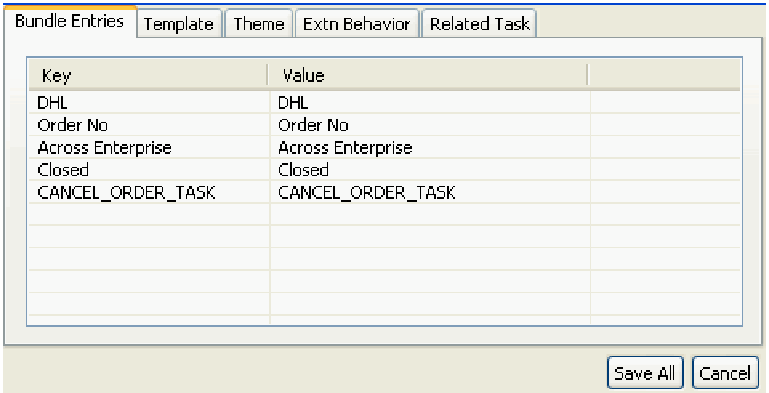


Table 6–1 Bundle Entries Tab

Field	Description
Key	Displays the control names of all the newly added fields for which the corresponding bundle entry does not exist. **Note:** If you add new related tasks, this field displays the related task identifier and the related task group identifier.
Value	Editable field. Enter the value associated with the key, if applicable. This value is used when retrieving the value for the keys from the bundle file. **Note:** For a related task and related task group, this field corresponds to the respective name.

3. Click Save All. The Save All window displays.

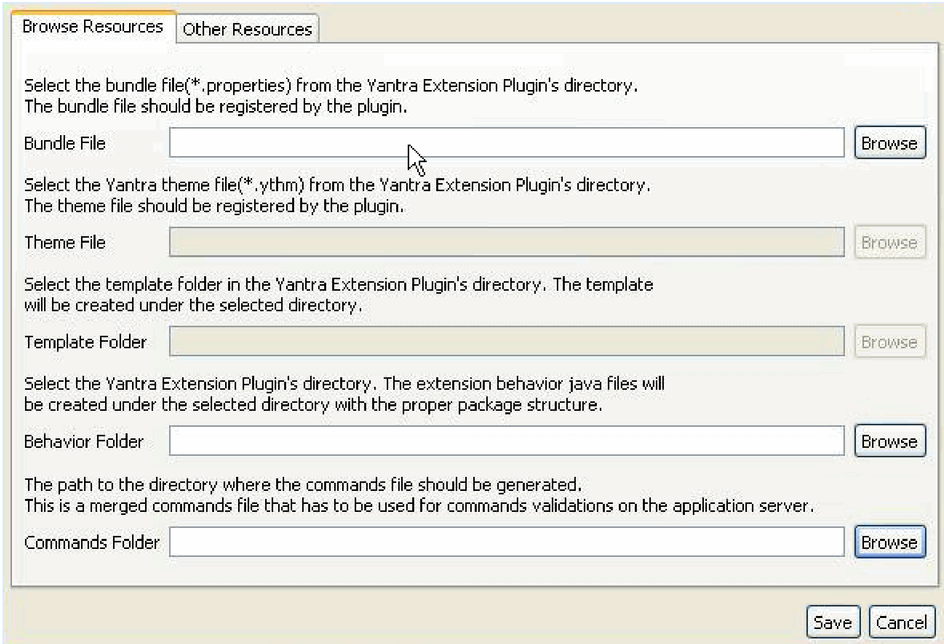


Table 6–2 Choose File for Saving Diffs

Field	Description
Browse Resources Tab This tab is used to save all the differences in the bundle and theme entries in their respective bundle and theme files. Also, the differences in the templates and behavior class in the respective template and behavior folders.	
Bundle File	Click Browse. The Save File dialog box displays. Select the bundle file in which you want to save the new bundle entries.
Theme File	Click Browse. The Save File dialog box displays. Select the theme file in which you want to save the new theme entries.
Template Folder	Click Browse. The Choose Folder dialog box displays. Select the template folder in which you want to save the new template.

Table 6–2 Choose File for Saving Diffs

Field	Description
Extn Behavior Folder	Click Browse. The Choose Folder dialog box displays. Select the folder in which you want to save the new extension behavior class.
Other Resources Tab Displays the path of the `plugin.xml` file in which all your related task changes are saved.	
Update Resource	Uncheck this box if you don't want to save the changes in the `plugin.xml` file.

> **Note:** Sterling Commerce recommends that you save all files, folders, and packages in the plug-in directory.

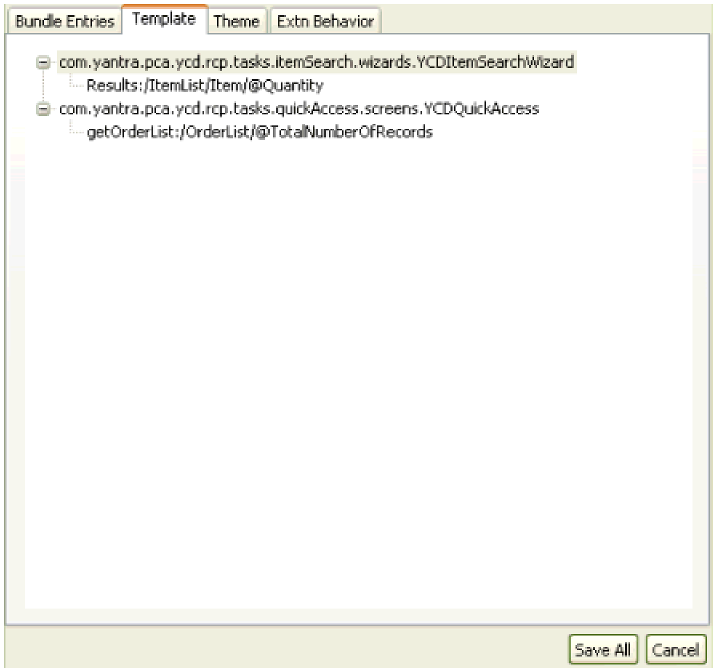
4. Click Save.
5. Click Close.

6.2 Synchronizing Templates

Whenever you specify bindings for newly added fields, you must synchronize the template folder with new templates that are defined for binding the newly added fields.

To synchronize the templates:

1. Start the Rich Client Platform Extensibility Tool.
2. Click . The Synchronize pop-up window displays. Select the Template tab. You can view the newly added bindings for the newly added fields under the respective form id.



3. Click Save All. The Choose File for Saving Diffs pop-up window displays. For field value descriptions, see Table 6–2.
4. Click Close.

6.3 Synchronizing Theme Files

Whenever you specify a new theme entry for newly added fields, you must synchronize the theme file.

To synchronize the theme file:

1. Start the Rich Client Platform Extensibility Tool.
2. Click . The Synchronize pop-up window displays. Select the Theme tab.

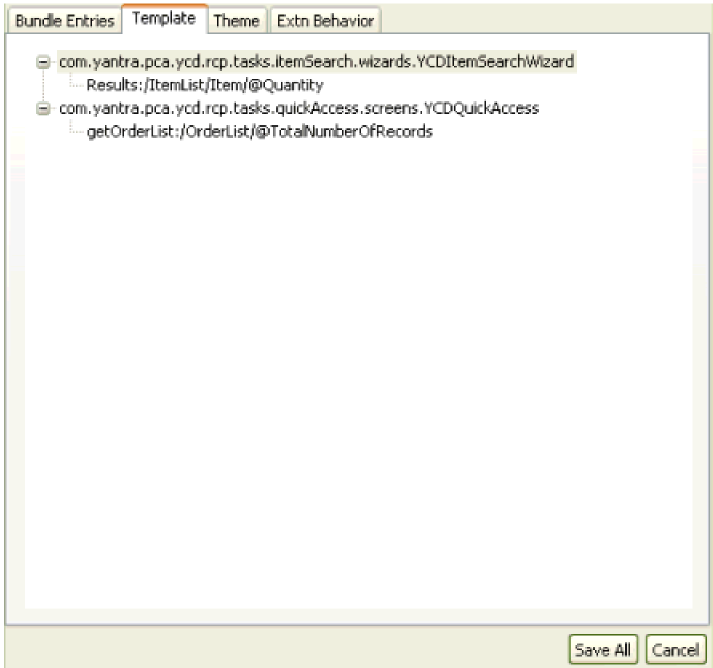


Table 6–3 Theme Tab

Field	Description
Themes	Displays the new theme entries.

3. Click Save All. The Choose File for Saving Diffs pop-up window displays. For field value descriptions, see Table 6–2.
4. Click Close.

6.4 Synchronizing Extension Behavior

When you create an extension behavior, you must synchronize the differences.

For more information about creating extension behavior, see Section 4.2, "Creating Extension Behavior".

There are two types of extension behavior:

- Normal Extension Behavior (YRCExtensionBehavior)—The system creates this behavior when you create an extension behavior for extending a Rich Client Platform screen that is not a wizard. When synchronizing a normal extension behavior, specify the name of the implementation class (extends YRCExtensionBehavior). You can add the logic to call an API or service or set the extension model in the init() method of the YRCExtensionBehavior class.
- Wizard Extension Behavior (YRCWizardExtensionBehavior)—The system creates this behavior when you create an extension behavior for extending a Rich Client Platform screen that is a wizard. When synchronizing the wizard extension behavior, specify the name of the implementation class (extends YRCWizardExtensionBehavior). You can add the wizard-specific logic to call an API or service or set the extension model in the initPage (String pageId) method of the YRCWizardExtensionBehavior class.

To synchronize extension behavior:

1. Start the Rich Client Platform Extensibility Tool.
2. Click . The Synchronize pop-up window displays. Select the Extn Behavior tab.

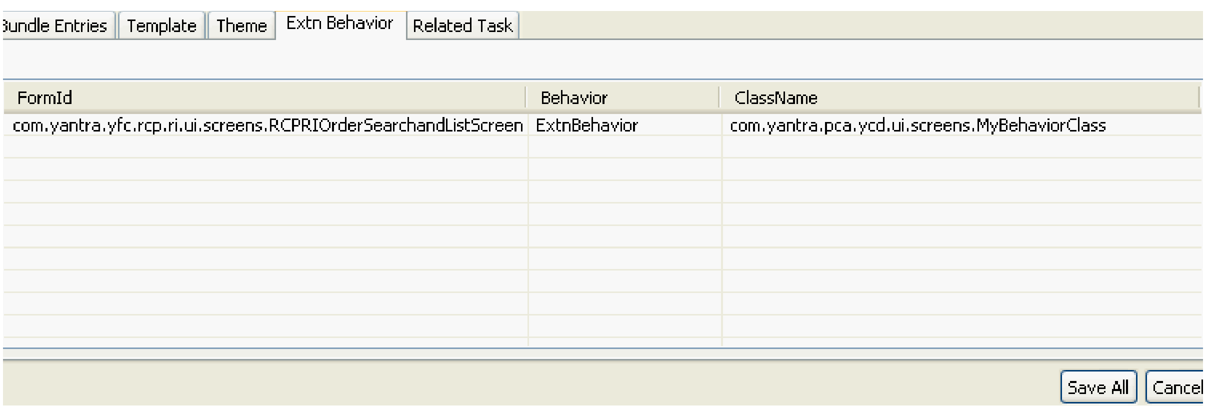


Table 6–4 Extn Behavior Tab

Field	Description
FormId	Displays the identifier of the form that you are extending.
Behavior	The type of behavior class automatically displays in the drop-down list. For example, WizardExtnBehavior or ExtnBehavior. • If you have created extension behavior for a Rich Client Platform screen that is a wizard, the drop-down list displays "WizardExtnBehavior". • If you have created extension behavior for a Rich Client Platform screen that is not a wizard, the drop-down list displays "ExtnBehavior".
ClassName	Enter the path of the behavior class. For example: `com.yantra.pca.ycd.ui.screens.MyBehaviorClass` where `com.yantra.pca.ycd.ui.screens` is the package name and MyBehaviorClass is the name of the behavior class.

3. Click Save All. The Choose File for Saving Diffs pop-up window displays. For field value descriptions, see Table 6–2.
4. Click Close. The extension behavior class is created in the package that you specified.
5. To retrieve the appropriate data for this newly added field or advanced column, write the code to call the required API or service, set the extension model, and setting the bindings for an advanced column in the newly created extension behavior class.

6.4.1 Calling APIs or Services

After synchronizing the extension behavior for a newly added field, you may want to retrieve data to populate the new field by calling an API or service.

6.4.2 Setting the Extension Model

You must set the extension model for the new namespaces that you define for the extended screens.

6.4.3 Setting Bindings for an Advanced Column

You must set bindings for the extended table and advanced column that you added using the Rich Client Platform Extensibility Tool.

6.5 Synchronizing Related Tasks

Whenever you specify bindings for the newly added fields, you must synchronize the template folder with new templates that are defined for binding the newly added fields.

To synchronize the templates:

1. Start the Rich Client Platform Extensibility Tool.
2. Click . The Synchronize pop-up window displays. Select the Template tab. You can view the newly added bindings for the newly added fields under the respective form id.

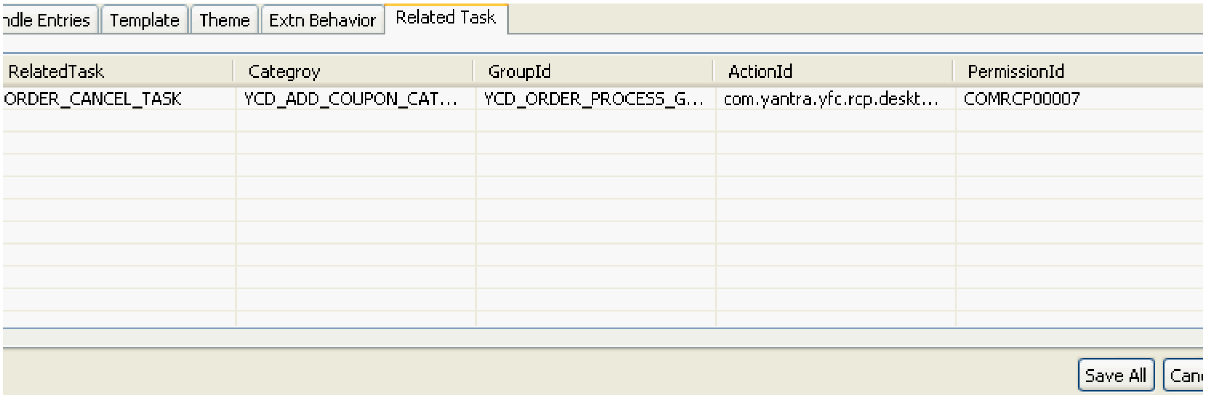


Table 6–5 Related Task Tab

Field	Description
RelatedTask	Displays the identifier of the new related task that you added.
Category	Displays the identifier of the identifier of the category in which the new related task is interested.
GroupId	Displays the identifier of the group in which you want to display the related task, so that the new related task can be displayed in the Related Tasks view.
ActionId	Displays the identifier of the action you want to invoke upon clicking the new related task.
PermissionId	Displays the resource identifier.

3. Click Save All. The Choose File for Saving Diffs pop-up window displays. For field value descriptions, see Table 6–2.
4. Click Close.

6.6 Displaying Hidden and Disabled Controls

You can show the hidden and disabled fields that are provided with Selling and Fulfillment Foundation using the Rich Client Platform Extensibility Tool.

To show the hidden and disabled fields:

1. Start the Rich Client Platform Extensibility Tool.

2. Click . The hidden and disabled fields display on the screen.

6.7 Viewing Shared Tasks

Using the Rich Client Platform Extensibility tool, you can view the shared task details used across applications or plug-ins. You can view the:

- Identifier of the shared task
- Name of the shared task
- A brief description of the shared task
- Template of the input XML model
- Template of the output XML model

To view the shared task details:

1. Start the Rich Client Platform Extensibility Tool.
2. Click View Shared Tasks and click once on the screen. The Shared Tasks window displays.

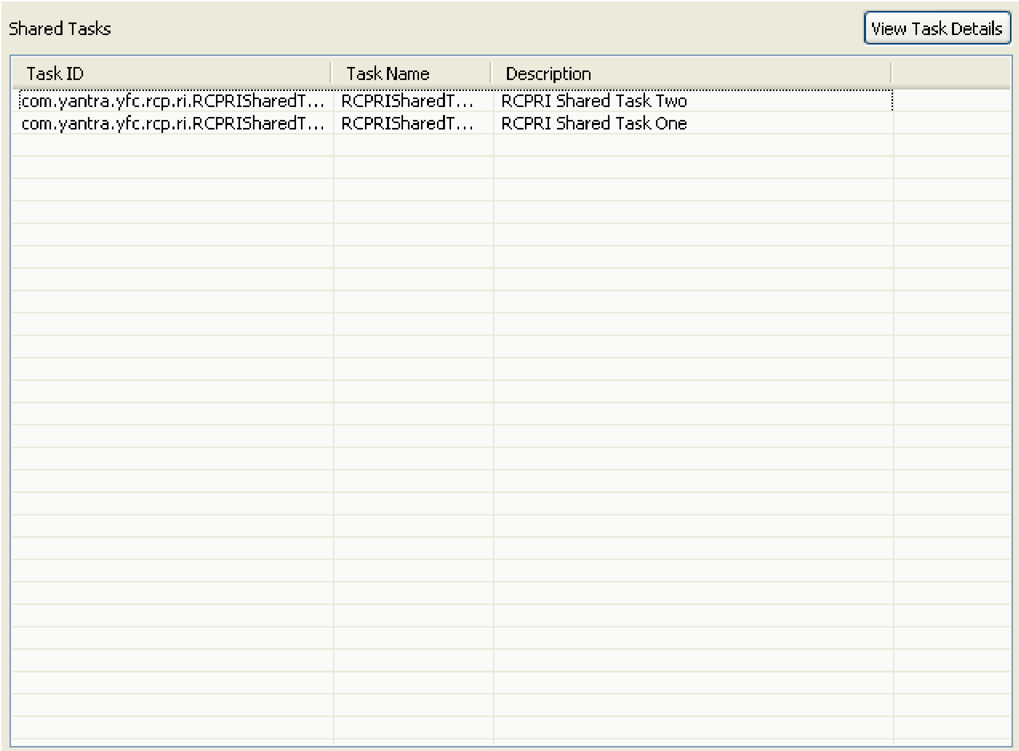


Table 6–6   Shared Tasks Window

Field	Description
Task ID	Displays the unique identifier of each shared task.
Task Name	Displays the name of each shared task.
Description	Displays the description of each shared task.

3. If you want to view more details about a particular shared task, select the shared task and click View Task Details. The Shared Task Details Panel pop-up window displays.

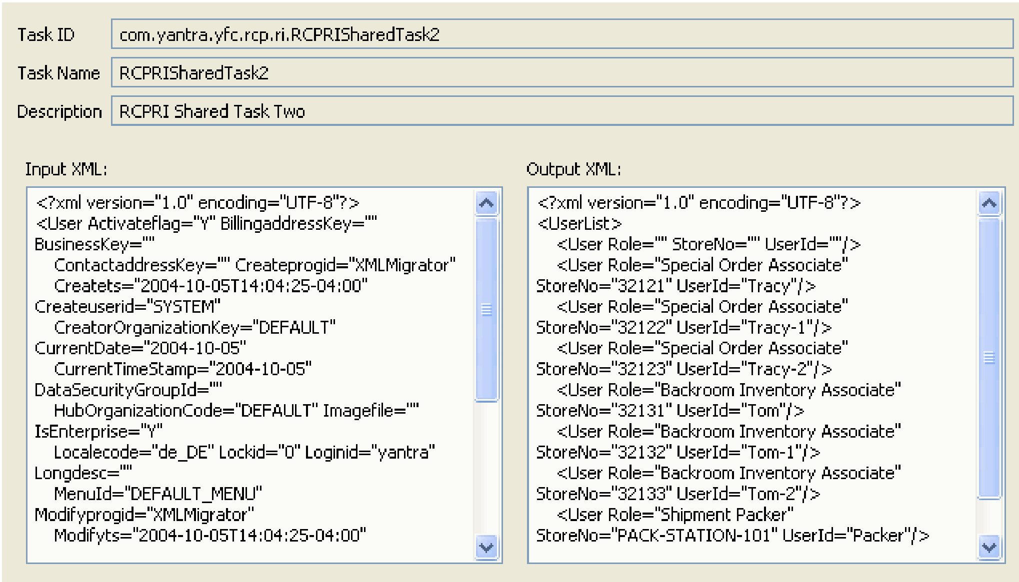


Table 6–7 Shared Task Details Panel Window

Field	**Description**
Task ID	Displays the unique identifier of the shared task.
Task Name	Displays the name of the shared task.
Description	Displays the description of the shared task.
Input XML	Displays the template structure that is used as an input XML by the shared task.
Output XML	Displays the template structure that is used as an output XML by the shared task.

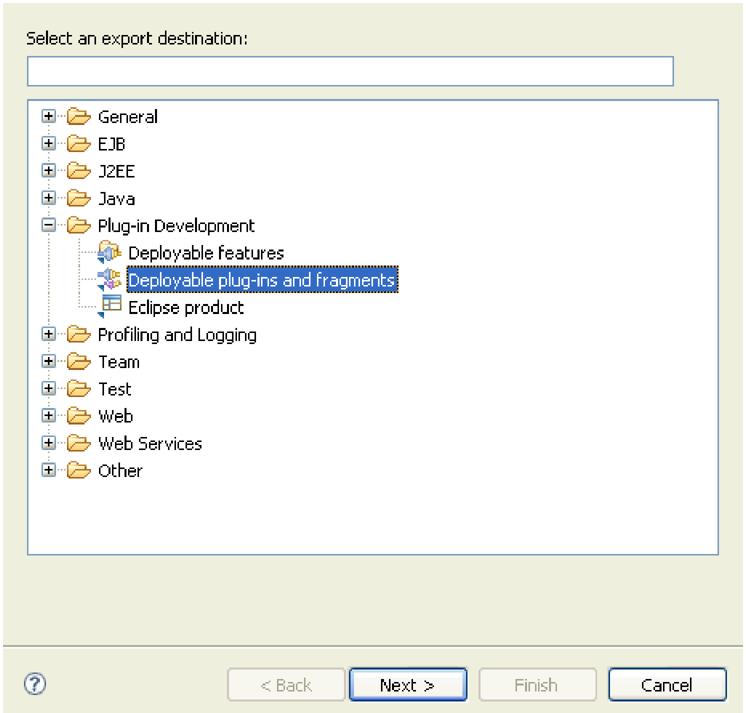
7

Building and Deploying Rich Client Platform Extensions

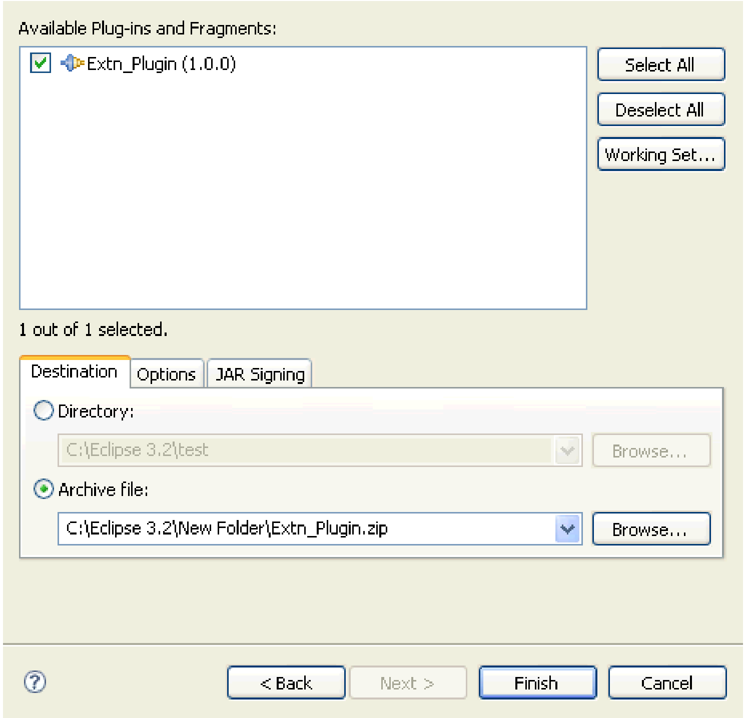
7.1 Building Rich Client Platform Extensions

Building the Rich Client Platform extensions is as follows:

1. Start the Eclipse SDK.
2. From the menu bar, select Window > Show View > Navigator. The plug-in project is displayed in the Navigator view.
3. Right-click on the plug-in project that you want to build and deploy.
4. Select Export... from the pop-up menu. The Export window displays.



5. From the list of export destinations, under Plug-in Deployment, select Deployable plug-ins and fragments.
6. Click Next.



7. In the Destination tab, Choose Archive file:.
8. Click Browse and browse to the folder where you want to store the exported plug-in zip file.
9. In the Options tab, make sure that the Package plug-ins as individual JAR archives box is checked.
10. Click Finish. The plug-in jar is generated and stored in the `plugins` folder in the zip file as specified in Step 8.

7.2 Deploying Rich Client Platform Extensions

After you build the Rich Client Platform extensions plugin jar, you must deploy this plug-in.

To deploy the Rich Client Platform extensions:

Copy the plugin jar that you built to the `plugins` directory of the <RCP_EXTN_FOLDER> folder and follow the steps as described in the "Deploying and Updating Rich Client Platform Application" chapter of the *Selling and Fulfillment Foundation: Installation Guide*.

Index

A

B

C

D

E

G

H

L

R

S

T